

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
DPTO. DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA



PROYECTO FIN DE CARRERA

**DESARROLLO DE PRÁCTICAS REMOTAS VIRTUALES DE
SEÑALES Y SISTEMAS MEDIANTE EASY JAVA SIMULATIONS**

AUTOR: CARLOS VILLA CARMONA

NOVIEMBRE 2009

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
DPTO. DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA



DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

**DESARROLLO DE PRÁCTICAS REMOTAS VIRTUALES DE
SEÑALES Y SISTEMAS MEDIANTE EASY JAVA SIMULATIONS**

PROYECTO FIN DE CARRERA
INGENIERÍA TÉCNICA INDUSTRIAL: ELÉCTRONICA INDUSTRIAL

AUTOR: **CARLOS VILLA CARMONA**
TUTOR: **RAMÓN BARBER CASTAÑO**

NOVIEMBRE 2009

A mis padres,
por su confianza en mí

A mis hermanos,
por su apoyo

A mis compañeros,
por todos esos momentos

Agradecimientos

En primer lugar quiero expresar mi profundo agradecimiento a todas aquellas personas que me han dado la oportunidad de desarrollarme, tanto intelectual como personalmente. Su influencia ha sido muy importancia para llegar al punto en el que me encuentro. No puedo nombrar a todos, pero si quiero reconocer el valor de algunos de ellos:

Mis padres y hermanos, por su apoyo y la confianza que han puesto en mí.

Ramón Barber, mi tutor del proyecto, de gran importancia en el desarrollo del mismo y que me ha aportado otro punto de vista del tema tratado a partir de su experiencia.

Por supuesto no me olvido de los amigos que aquí he hecho en estos años que han hecho que estos años hayan sido estupendos, hemos vivido momentos inolvidables y espero que lo sigamos viviendo mucho más tiempo.

Los profesores de la universidad, que han hecho posible alcanzar un nivel de conocimientos técnicos necesarios para la realización del proyecto.

Asimismo, a todas aquellas personas que habiendo colaborado de un modo u otro, por omisión involuntaria no han sido reflejadas en estas líneas.



Índice general

| | |
|---|-----------|
| CAPÍTULO 1: INTRODUCCIÓN | 1 |
| 1.1 PRESENTACIÓN | 2 |
| 1.2 INTRODUCCIÓN | 3 |
| 1.3 OBJETIVOS DEL PROYECTO | 4 |
| 1.4 SUMARIO DEL PROYECTO..... | 5 |
| CAPÍTULO 2: LABORATORIOS REMOTOS Y LABORATORIOS VIRTUALES..... | 6 |
| 2.1 INTRODUCCIÓN | 7 |
| 2.2 LABORATORIOS REMOTOS | 7 |
| 2.3 LABORATORIOS VIRTUALES..... | 12 |
| CAPÍTULO 3: EASY JAVA SIMULATIONS (EJS) | 18 |
| 3.1 ¿QUÉ ES EJS? | 19 |
| 3.2 EJECUTAR EJS..... | 20 |
| 3.2.1 Consola de EJS..... | 20 |
| 3.2.1.1 Opciones Básicas..... | 20 |
| 3.2.1.1.1 Espacio de trabajo | 21 |
| 3.2.1.2 Opciones Avanzadas..... | 23 |
| 3.2.1.3 Área de Mensajes..... | 25 |
| 3.2.2 Interfaz del usuario..... | 26 |
| 3.2.2.1 Descripción..... | 27 |
| 3.2.2.2 Modelo..... | 27 |
| 3.2.2.2.1 Variables..... | 28 |
| 3.2.2.2.2 Inicialización | 30 |
| 3.2.2.2.3 Evolución..... | 31 |
| 3.2.2.2.4 Relaciones fijas..... | 32 |
| 3.2.2.2.5 Código Propio..... | 33 |
| 3.2.2.3 Vista..... | 34 |
| CAPÍTULO 4: DEPÓSITO..... | 36 |
| 4.1 ENUNCIADO DEL PROBLEMA..... | 37 |



| | | |
|--|--|------------|
| 4.2 | RESOLUCIÓN DEL PROBLEMA | 38 |
| 4.2.1 | <i>Modelo no lineal</i> | 39 |
| 4.2.1.1 | Variables..... | 39 |
| 4.2.1.2 | Ecuaciones | 41 |
| 4.2.1.3 | Vista..... | 42 |
| 4.2.2 | <i>Modelo lineal</i> | 47 |
| 4.2.2.1 | Variables..... | 47 |
| 4.2.2.2 | Ecuaciones | 48 |
| 4.2.2.3 | Vista..... | 50 |
| CAPÍTULO 5: BARCO | | 52 |
| 5.1 | ENUNCIADO DEL PROBLEMA..... | 53 |
| 5.2 | RESOLUCIÓN DEL PROBLEMA | 54 |
| 5.2.1 | <i>Modelo no lineal</i> | 54 |
| 5.2.1.1 | Variables..... | 54 |
| 5.2.1.2 | Ecuaciones | 57 |
| 5.2.1.3 | Vista..... | 58 |
| 5.2.2 | <i>Modelo lineal</i> | 62 |
| 5.2.2.1 | Variables..... | 62 |
| 5.2.2.2 | Ecuaciones | 63 |
| 5.2.2.3 | Vista..... | 65 |
| CAPÍTULO 6: DEPÓSITO CONTROLADO..... | | 67 |
| 6.1 | ENUNCIADO DEL PROBLEMA..... | 68 |
| 6.1 | RESOLUCIÓN DEL PROBLEMA | 69 |
| 6.1.1 | Variables | 70 |
| 6.1.2 | Ecuaciones | 73 |
| 6.1.3 | Vista | 74 |
| CAPÍTULO 7 :CASOS PRÁCTICOS..... | | 79 |
| 7.1 | DEPÓSITO | 80 |
| 7.1.1 | <i>Incremento en el caudal de la entrada de 1 a 1,1 m³/s.....</i> | <i>80</i> |
| 7.1.2 | <i>Incremento en el caudal de la entrada de 1 a 4,5 m³/s.....</i> | <i>82</i> |
| 7.1.3 | <i>Incremento en el caudal de la entrada de 4 a 4,5 m³/s.....</i> | <i>84</i> |
| 7.2 | BARCO..... | 87 |
| 7.2.1 | <i>Masa 204 kg</i> | <i>87</i> |
| 7.2.2 | <i>Masa 304 kg</i> | <i>88</i> |
| 7.2.3 | <i>Masa 404 kg</i> | <i>90</i> |
| 7.3 | DEPÓSITO CONTROLADO..... | 92 |
| 7.3.1 | <i>Ejercicio 1: $h_{ref} = 0.149m$ y $K_p = 1000$.....</i> | <i>92</i> |
| 7.3.2 | <i>Ejercicio 2: $h_{ref} = 0.149m$ y $K_p = 500$.....</i> | <i>94</i> |
| 7.3.3 | <i>Ejercicio 3: $h_{ref} = 0.149m$ y $K_p = 1$.....</i> | <i>95</i> |
| 7.3.4 | <i>Ejercicio 4: $K_p = 1000$ y $h_{ref} = 0 - 0.2m$.....</i> | <i>96</i> |
| 7.3.5 | <i>Ejercicio 5: $K_p = 10$ y $h_{ref} = 0 - 0.2$.....</i> | <i>98</i> |
| CAPÍTULO 8: CONCLUSIONES Y TRABAJOS FUTUROS | | 99 |
| 8.1 | CONCLUSIONES | 100 |
| 8.2 | TRABAJOS FUTUROS..... | 101 |
| BIBLIOGRAFÍA..... | | 101 |



Índice de figuras

Capítulo 2: Laboratorios remotos y laboratorios virtuales

| | |
|---|----|
| FIGURA 1: ARQUITECTURA DE REFERENCIA PARA LA CONSTRUCCIÓN DE LABORATORIOS REMOTOS. | 9 |
| FIGURA 2: ASPECTO DE LA APLICACIÓN REMOTA DE UN LABORATORIO DE ACCESO REMOTO | 9 |
| FIGURA 3: MAQUETA DEL SISTEMA DE BOLA Y ARO EMPLEADA | 11 |
| FIGURA 4: ASPECTO DE LA APLICACIÓN PARA EL AJUSTE DEL CONTROLADOR DE POSICIÓN DEL ARO DE LA MAQUETA | 11 |
| FIGURA 5: ASPECTO DEL APPLLET JAVA QUE MUESTRA EL MOVIMIENTO DE UN COHETE | 15 |
| FIGURA 6: ASPECTO DEL DISEÑO DE LA VISTA CON EL ENTORNO EJS | 16 |
| FIGURA 7: ASPECTO DEL DISEÑO DEL MODELO CON EL ENTORNO EJS | 16 |
| FIGURA 8: LABORATORIO VIRTUAL CREADO CON EJS PARA DISEÑAR EL MOVIMIENTO DE UN PÉNDULO | 17 |

Capítulo 3: Easy Java Simulations (EJS)

| | |
|---|----|
| FIGURA 9: OPCIONES BÁSICAS | 21 |
| FIGURA 10: UNA ESTRUCTURA DE DIRECTORIOS TÍPICA. | 21 |
| FIGURA 11: OPCIONES AVANZADAS | 23 |
| FIGURA 12: ÁREA DE MENSAJES | 25 |
| FIGURA 13: INTERFAZ DEL USUARIO. DESCRIPCIÓN..... | 26 |
| FIGURA 14: PÁGINA DE EVOLUCIÓN..... | 31 |
| FIGURA 15: ELEMENTOS DE LA VISTA | 35 |

Capítulo 4: Depósito

| | |
|--|----|
| FIGURA 16: DEPÓSITO..... | 37 |
| FIGURA 17: VARIABLES NO LINEALIZADAS | 40 |
| FIGURA 18: VARIABLES COMUNES..... | 41 |
| FIGURA 19: ECUACIÓN EVOLUCIÓN | 41 |
| FIGURA 20: ECUACIÓN RELACIÓN FIJA | 42 |
| FIGURA 21: ELEMENTOS DE LA VISTA | 42 |



| | |
|---|----|
| FIGURA 22: PROPIEDADES DE LA ALTURA DEL AGUA..... | 44 |
| FIGURA 23: ELEMENTOS DE LA GRÁFICA | 45 |
| FIGURA 24: PROPIEDADES DEL BOTÓN PLAY | 46 |
| FIGURA 25: PROPIEDADES DEL BOTÓN PAUSE..... | 46 |
| FIGURA 26: PROPIEDADES DEL BOTÓN RESET | 46 |
| FIGURA 27: VARIABLES LINEALIZADAS | 47 |
| FIGURA 28: ECUACIÓN DIFERENCIAL..... | 50 |
| FIGURA 29: ECUACIONES RELACIONES FIJAS..... | 50 |
| FIGURA 30: PROPIEDADES DE LA ALTURA DEL AGUA..... | 51 |
| FIGURA 31: ELEMENTOS DE LA VISTA | 51 |
| FIGURA 32: SIMULACIÓN FINAL | 52 |

Capítulo 5: Barco

| | |
|---|----|
| FIGURA 33: ESQUEMA DEL BARCO | 53 |
| FIGURA 34: VARIABLES NO LINEALIZADAS | 56 |
| FIGURA 35: VARIABLES COMUNES..... | 56 |
| FIGURA 36: ECUACIONES EVOLUCIÓN..... | 57 |
| FIGURA 37: ECUACIÓN RELACIÓN FIJA | 57 |
| FIGURA 38: ELEMENTOS DE LA VISTA | 58 |
| FIGURA 39: PROPIEDADES DE LA ALTURA Y TAMAÑO DE LA CARGA..... | 59 |
| FIGURA 40: ELEMENTOS DE LA GRÁFICA | 60 |
| FIGURA 41: PROPIEDADES DEL BOTÓN PLAY | 61 |
| FIGURA 42: PROPIEDADES DEL BOTÓN PAUSE..... | 61 |
| FIGURA 43: PROPIEDADES DEL BOTÓN RESET | 61 |
| FIGURA 44: VARIABLES LINEALIZADAS | 63 |
| FIGURA 45: ECUACIÓN DIFERENCIAL..... | 64 |
| FIGURA 46: ECUACIONES RELACIONES FIJAS..... | 64 |
| FIGURA 47: PROPIEDADES DE LA ALTURA Y TAMAÑO DE LA CARGA..... | 65 |
| FIGURA 48: ELEMENTOS DE LA VISTA | 65 |
| FIGURA 49: SIMULACIÓN FINAL | 66 |

Capítulo 6: Depósito controlado

| | |
|--|----|
| FIGURA 50: DEPÓSITO..... | 68 |
| FIGURA 51: DIAGRAMA DE BLOQUES DEL REGULADOR..... | 68 |
| FIGURA 52: VARIABLES DEPÓSITO..... | 72 |
| FIGURA 53: VARIABLES AUXILIARES | 72 |
| FIGURA 54: ECUACIÓN EVOLUCIÓN | 73 |
| FIGURA 55: FUNCIÓN CÓDIGO PROPIO | 74 |
| FIGURA 56: FUNCIÓN RELACIÓN FIJA | 74 |
| FIGURA 57: ELEMENTOS DE LA VISTA I..... | 75 |
| FIGURA 58: ELEMENTOS DE LA VISTA II | 75 |
| FIGURA 59: PROPIEDADES DEL DEPÓSITO | 77 |
| FIGURA 60: PROPIEDADES DEL BOTÓN PLAY..... | 77 |
| FIGURA 61: PROPIEDADES DEL BOTÓN PAUSE..... | 78 |
| FIGURA 62: PROPIEDADES DE LA VENTANA GRAFICA Y EL BOTÓN SELECTOR | 78 |
| FIGURA 63: SIMULACIÓN FINAL | 79 |



Capítulo 7: Casos prácticos

| | |
|--|----|
| FIGURA 64: CAUDAL DE ENTRADA $1\text{ M}^3/\text{S}$ | 80 |
| FIGURA 65: CAUDAL DE ENTRADA $1,1\text{ M}^3/\text{S}$ | 81 |
| FIGURA 66: CAUDAL DE ENTRADA $1,1\text{ M}^3/\text{S}$ | 82 |
| FIGURA 67: CAUDAL DE ENTRADA $4,5\text{ M}^3/\text{S}$ | 83 |
| FIGURA 68: CAUDAL DE ENTRADA $4\text{ M}^3/\text{S}$ | 85 |
| FIGURA 69: CAUDAL DE ENTRADA $4,5\text{ M}^3/\text{S}$ | 86 |
| FIGURA 70: MASA DE 204 KG..... | 88 |
| FIGURA 71: MASA DE 304 KG..... | 90 |
| FIGURA 72: MASA DE 404 KG..... | 92 |
| FIGURA 73: $H_{REF} = 0,149\text{M}$ Y $K_P = 1000$ | 93 |
| FIGURA 74: $H_{REF} = 0,149\text{M}$ Y $K_P = 500$ | 94 |
| FIGURA 75: $H_{REF} = 0,149\text{M}$ Y $K_P = 1$ | 95 |
| FIGURA 76: $H_{REF} = 0\text{M}$ Y $K_P = 1000$ | 97 |
| FIGURA 77: $H_{REF} = 0,2\text{M}$ Y $K_P = 1000$ | 97 |
| FIGURA 78: $H_{REF} = 0\text{M}$ Y $K_P = 10$ | 99 |
| FIGURA 79: $H_{REF} = 0,2\text{M}$ Y $K_P = 10$ | 99 |



Capítulo 1

Introducción

1.1 Presentación

La titulación de Ingeniero Técnico Industrial: especialidad Electrónica impartida en la Universidad Carlos III de Madrid exige, para la obtención del título oficial, la calificación positiva de un Proyecto Final de Carrera, trabajo que ha de plasmar los conocimientos teórico-prácticos adquiridos durante los diferentes cursos académicos.

El Trabajo Fin de Carrera aquí presentado lleva por título: “*Desarrollo de prácticas remotas virtuales de Señales y Sistemas mediante Easy Java Simulations*”, ha sido realizado bajo la tutela del Tutor Ramón Barber Castaño perteneciente al Departamento de Ingeniería de Sistemas y Automática.

Se enmarca dentro del planteamiento de desarrollo e implantación de laboratorios virtuales.

El proyecto de construcción de éstos laboratorios virtuales se ha desarrollado con la herramienta Easy Java Simulations (simulaciones sencillas en Java), también conocido como EJS o Ejs, es una herramienta de autor creada en Java que ayuda a no programadores a crear simulaciones interactivas en Java, habitualmente con fines de enseñanza o aprendizaje.

La elección de Java como lenguaje de desarrollo se justifica por su gran aceptación por la comunidad internacional de Internet y por el hecho de que está soportado por prácticamente todas las plataformas de software. Esto significa que EJS, y las simulaciones creadas con él, pueden usarse como programas independientes virtualmente en todos los sistemas operativos, o ser distribuidas por Internet y ejecutadas en páginas Web por los navegadores más conocidos (siempre que tengan instalado Java) [2].

1.2 Introducción

Es sabido que la experimentación obliga a los alumnos a implicarse en el aprendizaje convirtiéndose en una parte esencial del aprendizaje de la mayoría de las ramas científicas y técnicas. Existe un gran número de estudios de psicología cognitiva que demuestran que las personas adquieren mejor el conocimiento haciendo cosas y reflexionando sobre las consecuencias de sus acciones que mirando o escuchando a alguien que les cuenta lo que deben aprender [1].

Sin embargo, la realización de experimentos con plantas reales suele ser costosa en lo que se refiere a tiempo, dinero y energía, ya que requiere la puesta a punto de unas infraestructuras docentes normalmente caras que son difíciles de mantener en buenas condiciones. Además, es frecuente que una vez que estas infraestructuras sean puestas a punto permanezcan infrautilizadas debido fundamentalmente a la limitación en el tiempo que pueden ser utilizadas por parte de los alumnos ya que se suele tratar de infraestructuras sensibles a usos indebidos que éstos les puedan dar dificultándose la realización de las necesarias prácticas con equipos reales [4].

Por otra parte, la misión de los educadores debe ser no sólo formar en las materias que imparten, sino conseguir que los alumnos sean capaces de manejar las herramientas disponibles para resolver los problemas con los que se encontrarán en su vida profesional. Evidentemente, entre estas herramientas cabe señalar las TIC (Tecnologías de la Información y las Comunicaciones).

Es precisamente de la mano de las TICs desde donde se pueden aportar algunas soluciones para ampliar el acceso a la experimentación en forma de laboratorios remotos y virtuales. Así se conseguirán simultáneamente dos objetivos didácticos: realizar prácticas relacionadas con la asignatura ampliando la disponibilidad de los laboratorios y formar a nuestros alumnos en el uso de las TICs [1].

1.3 Objetivos del proyecto

El objetivo principal es el desarrollo de prácticas de la asignatura Señales y Sistemas sobre sistemas simulados que corren de forma remota en un servidor docente. Con dichas prácticas se pretende la correcta comprensión de los fenómenos simulados por parte del estudiante, que le permite la realización no presencial de prácticas de laboratorio.

En éste caso las simulaciones se tratan de un depósito en lazo abierto, un barco en lazo abierto y un depósito controlado en lazo cerrado. En el caso de los sistemas de lazo abierto se estudiará como se estabilizan y qué tipo de respuesta ofrecen en su linealización. Sin embargo en el sistema de lazo cerrado se verá como el sistema se regula automáticamente para una entrada ofrecida por el usuario.

Los objetivos, pueden concretarse en:

- Estudio de Easy Java Simulation como herramienta de diseño de Sistemas Remotos Virtuales.
- Desarrollo de tres modelos con Easy Java Simulations: un depósito en lazo abierto, un barco en lazo abierto y un depósito controlado.
- Desarrollo de tres prácticas: dos destinadas a comparar los sistemas linealizados con los reales y uno para estudiar el regulador proporcional.

De este modo se pretende que los estudiantes mediante dichas simulaciones realicen las prácticas de Señales y Sistemas, involucrándose así en lo que se denomina por algunos investigadores educacionales *modelado constructivo* [2]. Este enfoque presenta la ventaja de que los estudiantes hagan ciencia de una manera exploratoria y constructiva, cumpliéndose así varias de las prácticas recomendadas para el trabajo en clase.

1.4 Sumario del proyecto

El Proyecto Final de Carrera se divide en los nueve apartados que se muestran en el índice que figura al inicio. A continuación se realiza un análisis de cada uno de ellos con objeto de sintetizar en pocas líneas la totalidad del mismo.

El capítulo 2 del presente proyecto esta destinado a la presentación de los laboratorios remotos y los laboratorios virtuales. Se ofrece una información detallada para explicar en qué consiste cada uno de estos tipos de laboratorios.

El capítulo 3 muestra un resumen de uso de la herramienta de trabajo EJS. Se detalla el proceso que consiste en crear un nuevo proyecto, en elaborar una simulación y ejecutarla.

En el capítulo 4 se describe de forma detallada el modo en que se lleva a cabo la simulación del depósito, comentando cada paso tanto del modelo lineal como el no lineal (modelo real).

En el capítulo 5 se describe de forma detallada el modo en que se lleva a cabo la simulación del barco, comentando cada paso tanto del modelo lineal como el no lineal (modelo real).

En el capítulo 6 se describe de forma detallada el modo en que se lleva a cabo la simulación del depósito controlado, comentando cada paso tanto del modelo lineal como el no lineal (modelo real).

El capítulo 7 muestra los resultados en los que modificando las variables de los tres sistemas, se obtienen distintas respuestas para cada uno de ellos. Una vez obtenidos los resultados se aportan una serie de conclusiones para explicar el funcionamiento.



El capítulo 8 contiene las conclusiones del proyecto tras haber desarrollado todo lo que se refiere a los laboratorios remotos y virtuales.

Finalmente se incluye la bibliografía donde se encuentran las referencias los libros y sitios Web de interés.



Capítulo 2

Laboratorios remotos y laboratorios virtuales

2.1 Introducción

Sebastián Dormido [3] clasifica los laboratorios en función de dos criterios: la forma de acceder a los recursos (local o remota) para propósitos de experimentación y la naturaleza del sistema físico (real o virtual), con lo que los entornos de experimentación quedarían clasificados de acuerdo a la siguiente tabla:

| | REAL | VIRTUAL |
|---------------|---|--|
| Local | Laboratorios presenciales con plantas reales | Laboratorios presenciales con plantas simuladas |
| Remoto | Teleoperación de una planta real | Laboratorio remoto con plantas simuladas |

Tabla 1: clasificación de los entornos experimentales

A pesar de que nada puede compararse con la interacción real con las plantas físicas (laboratorios reales y locales), también existen otras opciones que proporcionan a los alumnos la impresión de que están interactuando con plantas reales. Así, normalmente se entiende por laboratorio remoto aquél que existe y puede ser manipulado de forma remota a través de Internet, haciendo uso de Webcams, hardware específico para la adquisición local de datos y software para dar una sensación de proximidad con el equipamiento y por laboratorio virtual (local o remoto) la acción de recrear el comportamiento de plantas de experimentación, mediante software, que sólo existen en ordenadores.

2.2 Laboratorios remotos

La interacción directa con el equipamiento del laboratorio aporta una experiencia difícil de igualar dado que, además de las variables medidas, los alumnos

perciben los experimentos con los cinco sentidos (vista, tacto, oído, olfato e incluso, a veces, gusto) [5]. También resulta de alto interés didáctico la resolución de problemáticas asociadas a la puesta a punto de los equipos, la configuración de los experimentos y la medida de las variables de interés, etc. Sin embargo, tal y como se comentó en la introducción, es muy frecuente que los laboratorios permanezcan infrautilizados debido a diversas razones (por ejemplo, horario limitado de acceso al laboratorio, ausencia de personal encargado o mantenimiento del equipamiento en estado inadecuado). En esta situación, el acceso remoto restringido a un equipamiento real puede consistir en una solución muy interesante si se consigue paliar la ausencia del contacto directo con el equipamiento con tecnología: Webcams, micrófonos, hardware de adquisición de datos, etc. En definitiva, todos aquellos recursos que permitan adquirir información ambiental que pueda ser enviada a través de redes informáticas de tipo Internet y reproducida de forma remota.

Por el contrario, la construcción de laboratorios remotos, se convierte en una tarea más compleja que la implementación de laboratorios presenciales. Por supuesto, permanece la problemática habitual de los laboratorios presenciales que incluye tanto el diseño de los experimentos como la configuración del equipamiento empleado para permitir realizar el experimento, pero además, se debe construir una infraestructura de acceso remoto que permita integrar la información obtenida a partir de diferentes medios y presentarla a los alumnos dentro de una plataforma sencilla, multiplataforma y fácil de configurar como pueden ser los navegadores Web, proporcionando unos parámetros aceptables de calidad de servicio [6].

A continuación se presenta un enfoque basado en tecnologías estándar (WWW, lenguaje Java, tecnologías orientadas a objetos distribuidas, etc.) con la intención de proporcionar un marco genérico que se adapte a un gran número de situaciones. Dicha metodología proporciona una arquitectura de referencia, mostrada en la figura 1, así como unas pautas a seguir para completar los diferentes componentes involucrados en los laboratorios remotos (Servidor de Aplicaciones, Aplicaciones Remotas, etc.) de

forma que los diseñadores se puedan concentrar en la funcionalidad del laboratorio remoto.



Figura 1: arquitectura de referencia para la construcción de laboratorios remotos

Como puede observarse en la figura 2 el laboratorio remoto se integra en un navegador Web lo que facilita la sencillez de uso por parte de los alumnos. En dicha figura se muestra el aspecto de la aplicación remota para un perfil de usuario. Como puede observarse en la figura, se combinan diferentes tipos de información visual (vídeo, e información textual y numérica), así como botones con los que los alumnos pueden ejercer acciones sobre la célula [3].

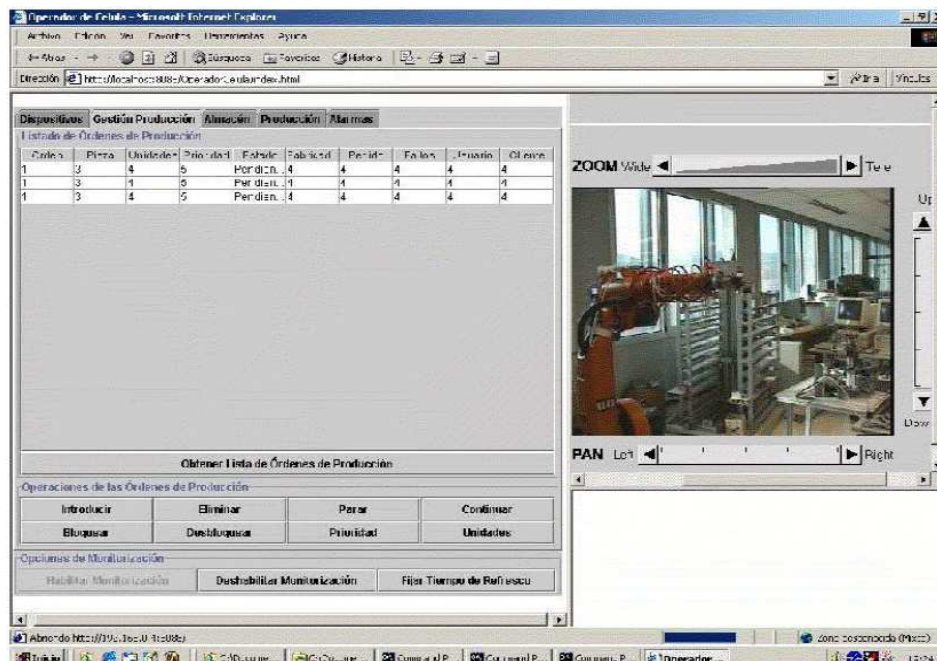


Figura 2: aspecto de la aplicación remota de un laboratorio de acceso remoto



Esta metodología se mostró adecuada para aplicaciones de alta complejidad, sin embargo, en muchas ocasiones se precisan implementar prácticas más sencillas que no requieran la intervención de personal altamente cualificado. En esta línea existen una serie de paquetes comerciales. Entre ellos merece la pena mencionar MatLab y LabVIEW. Se trata de dos paquetes software ampliamente utilizados tanto a nivel académico como profesional que proporcionan un gran número de módulos que facilitan el trabajo en diferentes entornos técnicos y científicos [3].

Cabe mencionar que aunque en sus orígenes MatLab (Matrix Laboratory) se desarrolló como un paquete software optimizado para realizar cálculos con vectores y matrices, ha ido evolucionando para irse adaptando a diferentes ramas de ingeniería y la ciencia añadiendo módulos adecuadas para diferentes áreas. En la actualidad, entre otras muchas, hay dos especialmente interesantes para crear laboratorios remotos. Una de ellas (Real Time Workshop) permite generar el código para sistemas embebidos de adquisición de datos y control que interaccionen con plantas reales y la otra (MatLab Web Server) permite enviar datos a través de tráfico WWW a un servidor de MatLab para que sean procesados y devuelve los resultados para que sean visualizados por un navegador Web.

Por su parte, LabVIEW es un paquete que originariamente fue desarrollado para la creación de aplicaciones de test, control y adquisición de datos. Sin embargo ha sido enriquecido con una multitud de librerías y módulos que implementan funciones complejas evolucionando en la actualidad de forma casi paralela a MatLab. LabVIEW no requiere unos conocimientos avanzados de programación y es ampliamente utilizado por la comunidad científica y técnica. Además, dispone de una gran cantidad de drivers para dispositivos de adquisición de datos y control.

Otro ejemplo es el de un laboratorio remoto para un curso básico en ingeniería de control que intenta maximizar el uso de una única planta, un sistema de bola y aro (Ver figura 3).

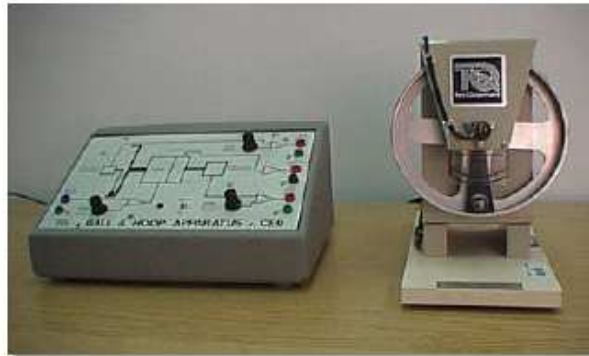


Figura 3: maqueta del sistema de bola y aro empleada

En este caso, se presentan varios experimentos con diferentes grados de dificultad, que incluyen el modelado experimental de un motor DC, su control de posición y velocidad, el estudio de la resonancia del sistema de bola y aro, el estudio de sistemas de fase no-mínima y las analogías entre la planta y problemas de control de la vida real. En la figura 4 puede observarse el entorno de usuario para el control de posición del sistema. Evidentemente, de forma paralela a esta aplicación se utilizan Webcams y micrófonos/altavoces para que los usuarios remotos puedan percibir información del laboratorio.

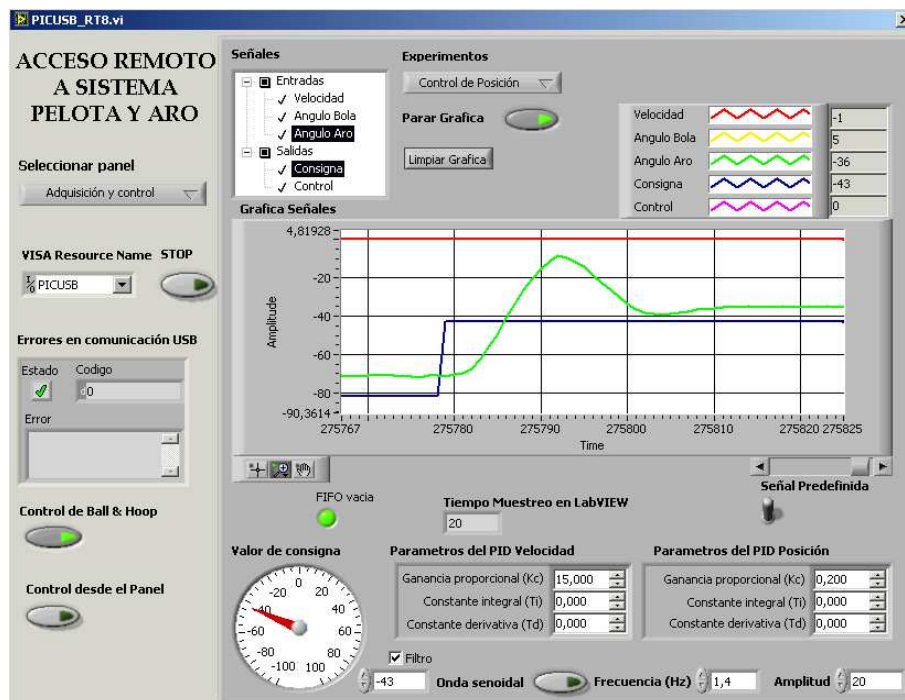


Figura 4: aspecto de la aplicación para el ajuste del controlador de posición del aro de la maqueta

2.3 Laboratorios virtuales

Como ya se indicó en la introducción, como alternativa a los laboratorios presenciales y remotos se pueden utilizar laboratorios virtuales. En este caso se usan los ordenadores para simular el comportamiento de los sistemas a estudiar haciendo uso de modelos matemáticos [3]. Aunque en este caso no se interacciona con plantas reales, la experimentación con modelos simulados es comparable siempre que se cumplan las siguientes premisas: se usen modelos matemáticos realistas que representen al alumno los detalles importantes del sistema a analizar y se complementen las gráficas que muestran la evolución temporal de los sistemas con animaciones que permitan a los alumnos visualizar y entender mejor el comportamiento del sistema.

La creación de laboratorios virtuales tiene algunas ventajas importantes con respecto a los laboratorios reales (presenciales o remotos). Dado que un laboratorio virtual se basa en modelos matemáticos que se ejecutan en ordenadores, su configuración y puesta a punto es mucho más sencilla que la configuración y puesta a punto de los laboratorios reales. Además, presentan un grado de robustez y seguridad mucho más elevado ya que al no haber dispositivos reales éstos no pueden causar problemas en el entorno [4]. Sin embargo, como inconveniente con respecto a los laboratorios reales cabe señalar que los laboratorios virtuales están limitados por el modelo y para poder ser manejables éstos tienden a simplificarse, con lo que se pierde información con respecto al sistema real. Además, la experimentación con sistemas reales (aún siendo de forma remota) siempre es un valor añadido para los alumnos [3].

Una forma interesante de utilizar los laboratorios virtuales es junto a los laboratorios reales (presenciales o remotos), de forma que los alumnos realizarían primero las prácticas en laboratorios virtuales, para pasar posteriormente, cuando el instructor lo considerase oportuno, al laboratorio real. Así se consiguen varios objetivos importantes como son [3]:



1. Familiarizarse con el experimento: evitando que los estudiantes puedan acudir al aula sin haber realizado trabajo previo.

2. Optimizar el uso de los recursos: los estudiantes requieren menos tiempo para realizar las prácticas, haciéndose un mejor uso de los laboratorios reales, tanto locales como remotos.

3. Disminución del uso incorrecto del equipamiento: frecuentemente los dispositivos utilizados en laboratorios reales son delicados, lo que se acentúa si se les hace trabajar fuera de las condiciones de trabajo para las que están diseñados.

4. Comparación del comportamiento de modelos matemáticos frente a dispositivos reales: los modelos matemáticos se obtienen simplificando el comportamiento de los dispositivos reales, lo que puede producir comportamientos sensiblemente diferentes. Al tener la oportunidad de comparar ambos comportamientos, los alumnos pueden extraer conclusiones acerca de la importancia del modelado realizado.

5. Formar en metodologías de trabajo: en la futura vida laboral del estudiante habitualmente se construyen primero modelos matemáticos de los sistemas que simularán bajo diferentes circunstancias como paso previo a construir prototipos, mucho más caros, con los que experimentar.

6. Manejo de herramientas informáticas actuales: en la vida profesional, e incluso en la vida diaria, la destreza en el uso de las herramientas informáticas, sean del ámbito que sean, es un elemento diferenciador. Con ello se consigue aportar al alumno una serie de conocimientos transversales que si bien pueden no ser el objetivo principal del laboratorio que se esté diseñando, le servirán en muchos ámbitos en el futuro.

7. Repetitividad de los experimentos: dado que el comportamiento de los sistemas a estudiar se obtiene mediante el modelado matemático de la realidad, los



alumnos pueden repetir de forma totalmente correcta las condiciones bajo las que se realizaron los experimentos y reproducirlos ante el docente en caso de necesidad, con la seguridad de que el resultado será el mismo que ellos vieron en su momento.

8. Multiplicidad de experimentos simultáneos: como el único recurso necesario para este tipo de laboratorios es un ordenador, potencialmente todos y cada uno de los alumnos podría estar realizando simultáneamente su experimento sin interferir con sus compañeros, eliminando la necesaria secuencialidad que se da en los laboratorios reales, ya sean locales o remotos. Además de la ventaja que supone en tiempo, se favorecen procesos colaborativos como el de “tormenta de ideas”, ya que todos y cada uno de los alumnos ha realizado su experimento y puede aportar su percepción de lo que allí ha ocurrido [6].

Los laboratorios virtuales pueden ser locales o remotos [3]. La diferencia reside en el lugar donde se realizan las simulaciones, que pueden ser, en el caso de laboratorios virtuales remotos, un servidor lejano que ejecuta los cálculos, o de forma local, en el caso de los laboratorios virtuales locales, donde se ejecutan los recursos haciendo uso de la potencia de cálculo de los alumnos.

La figura 5 muestra una imagen de un experimento virtual para describir el movimiento de un cohete. Previamente al experimento los alumnos acceden a una explicación del mismo. A pesar de tratarse de un excelente recurso para el autoaprendizaje de la física, cabe señalar que se podría aumentar aún más el grado de interacción con los experimentos virtuales. Por otra parte, los experimentos se han mantenido, en general, simples, con la evidente intención de que los alumnos asimilen mejor los conceptos, sin embargo, los alumnos también deben ser educados en la complejidad para poder afrontar mejor su futura vida profesional.

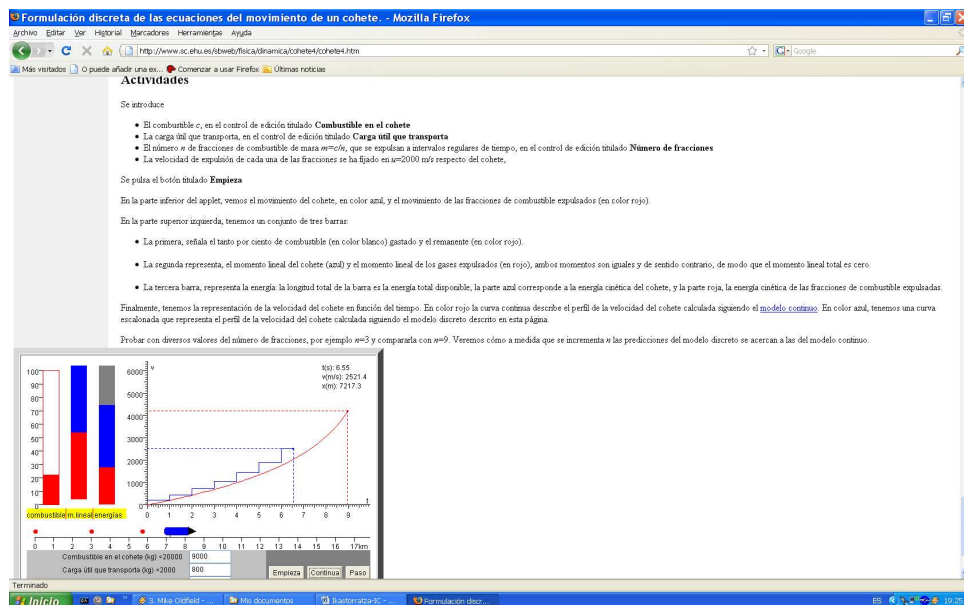


Figura 5: aspecto del applet Java que muestra el movimiento de un cohete

Otra iniciativa que vale la pena destacar es Easy Java Simulations, EJS. Se trata de un entorno de simulación dinámica, sencillo de utilizar, de uso libre que ha sido desarrollado por el profesor F. Esquembre y que aunque inicialmente estaba orientado a la enseñanza de la física ha ido extendiéndose su uso a otros entornos para construir laboratorios virtuales. El entorno se basa en el paradigma MVC (Modelo – Vista – Controlador) de forma que el diseño se divide en tres partes:

- **Modelo:** se encarga de la lógica del sistema (el modelo matemático).
- **Vista:** se encarga del gestionar y mostrar interfaz de usuario.
- **Controlador:** define las acciones que el usuario puede realizar sobre el modelo.

Las figuras 6 y 7 muestran el aspecto de los entornos de diseño de la vista y del modelo:



Figura 6: aspecto del diseño de la vista con el entorno EJS

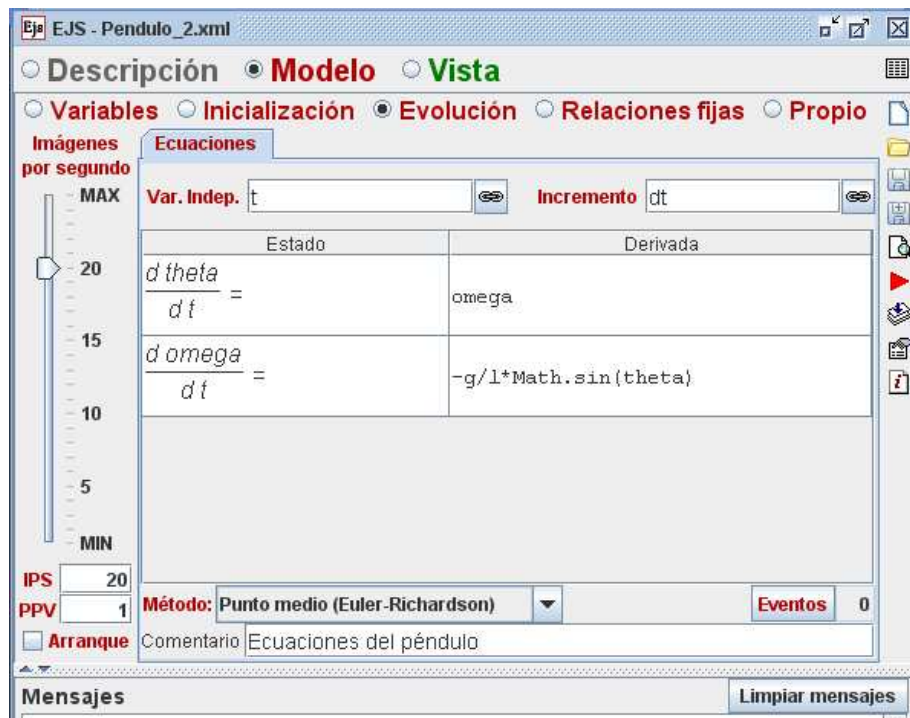


Figura 7: aspecto del diseño del Modelo con el entorno EJS

Como puede observarse en la figura 6, para diseñar la vista se van añadiendo una serie de elementos con los que se construirá el laboratorio virtual, mientras que en la

figura 7 puede observarse cómo se ha implementado el modelo matemático para que se ejecute en el laboratorio virtual.

Una vez diseñado el laboratorio, se creará un archivo *.jar* con el programa compilado en Java que ejecutará el laboratorio virtual, siendo por tanto el único requisito que los alumnos tengan la máquina virtual de Java instalada. En la figura 8 puede observarse el aspecto de los laboratorios virtuales creados con EJS, concretamente en este caso se ha diseñado un laboratorio virtual para diseñar el movimiento que realiza un péndulo para distintos tipos de masas y diferentes inercias. Se trata de un problema típico en el que los alumnos tienen la posibilidad de interactuar con el sistema cambiando determinados parámetros y observar cuál es el efecto de los cambios en su evolución temporal.

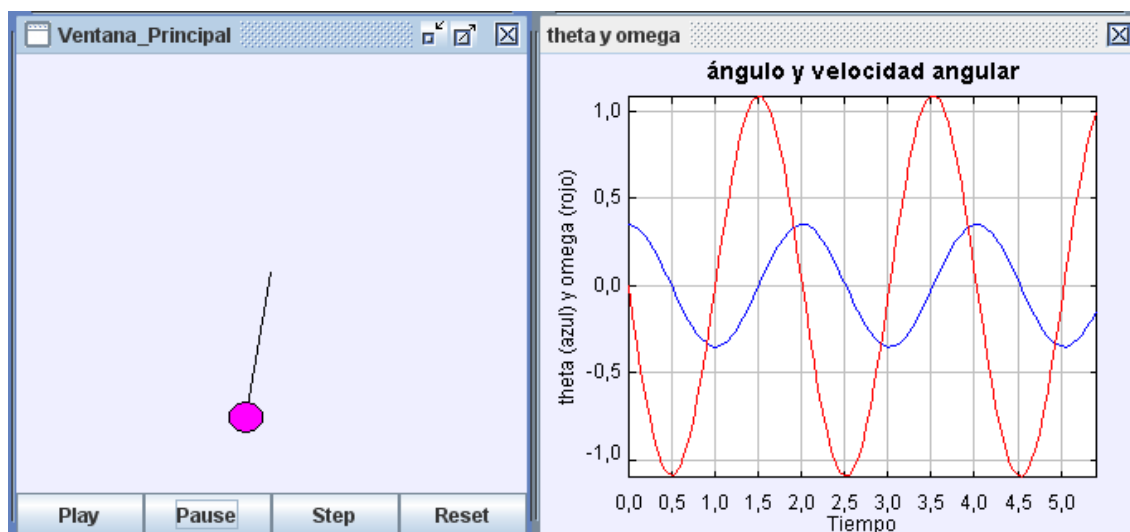


Figura 8: laboratorio virtual creado con EJS para diseñar el movimiento de un péndulo

También existen otras alternativas, basadas en software de cálculo como, por ejemplo, MatLab o Scilab. Ambas herramientas proporcionan entornos gráficos con los que se pueden construir animaciones, aunque a diferencia de los laboratorios virtuales creados con EJS o directamente en applets Java, en este caso se requiere que los alumnos se instalen dicho software, lo que dificulta su uso. Además, de que por ejemplo, en el caso de MatLab, se trata de un software de coste elevado. Una forma de



evitar este inconveniente en el caso de MatLab, es utilizando laboratorios virtuales remotos de forma que haciendo uso de MatLab Web Server, los alumnos sólo precisan de un navegador con el que se conectarán a un servidor que ejecutará el motor de MatLab donde se realizarán las simulaciones y que preparará las páginas Web dinámicas que verá el cliente.



Capítulo 3

Easy Java Simulations (EJS)

3.1 ¿Qué es EJS?

Easy Java Simulations es una herramienta de software diseñada para la creación de simulaciones discretas por computador [2].

Una simulación discreta por computador, o simplemente una simulación por computador, es un programa de computador que intenta reproducir, con fines pedagógicos o científicos, un fenómeno natural a través de la visualización de los diferentes estados que éste puede presentar. Cada uno de estos estados está descrito por un conjunto de variables que cambia en el tiempo debido a la iteración de un cierto algoritmo.

Todo esto significa que EJS es un programa que le ayuda a crear otros programas; más precisamente, simulaciones científicas.

Existen muchos programas que ayudan a crear otros programas. Lo que hace a EJS diferente de la mayoría de los otros productos es que EJS no ha sido diseñado para hacer la vida más sencilla a los programadores profesionales, sino que ha sido concebido por profesores de ciencias para ser usado por profesores y estudiantes de ciencias. Esto es, para personas que están más interesadas en el contenido de la simulación, en el fenómeno mismo que se simula, que en los aspectos técnicos necesarios para construir la simulación [2].

Easy Java Simulations es una herramienta de modelado y de autor expresamente dedicada a esta tarea. Ha sido diseñado para permitir a sus usuarios trabajar a un alto nivel conceptual, usando un conjunto de herramientas simplificadas y concentrando la mayoría de su tiempo en los aspectos científicos de la simulación, y pidiendo al computador que realice automáticamente todas las otras tareas necesarias pero fácilmente automatizables.

No obstante, el resultado final, generado automáticamente por EJS a partir de la descripción del autor, podría pasar, en términos de eficiencia y sofisticación, por la creación de un programador profesional.

En particular, EJS crea aplicaciones Java que son independientes y multiplataforma, o *applets* que se pueden visualizar usando cualquier navegador Web (y por tanto ser distribuidos a través de Internet), que pueden leer datos a través de la red y ser controlados usando *scripts* (conjuntos de instrucciones) incluidos en las páginas HTML.

Dado que existe un valor educativo añadido en el proceso de creación de una simulación, EJS puede ser usado también como una herramienta pedagógica.

3.2 Ejecutar EJS

Al arrancar EJS, se abrirán dos ventanas, la consola de EJS y la interfaz de usuario.

3.2.1 Consola de EJS

La primera ventana es la consola de EJS, que presenta tres pestañas: “Opciones Básicas”, “Opciones Avanzadas” y “Área de Mensajes”.

3.2.1.1 Opciones Básicas

La pestaña de Opciones Básicas de la consola de EJS (ver figura 9) contiene información sobre:

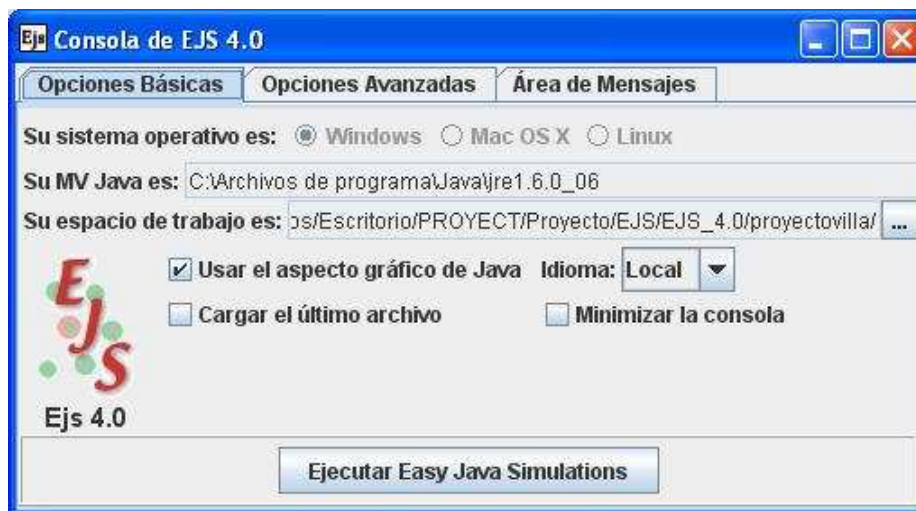


Figura 9: opciones básicas

- 1.- La dirección en el disco duro de la MV (Máquina Virtual) de Java.
- 2.- El espacio de trabajo actual.

3.2.1.1.1 Espacio de trabajo

EJS utiliza el concepto de un “espacio de trabajo” para organizar su trabajo. Un espacio de trabajo es un directorio donde EJS almacena todos sus ficheros de simulación para un determinado proyecto. Un espacio de trabajo puede contener un número ilimitado de simulaciones.

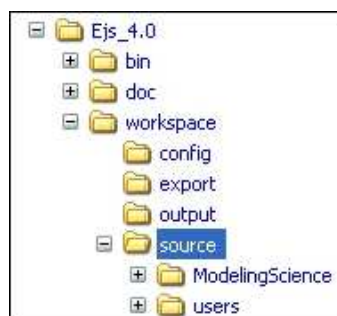


Figura 10: una estructura de directorios típica.



Dentro de cada directorio que se configure como espacio de trabajo, se encontrarán cuatro subdirectorios:

- “config”: Este subdirectorio contiene ficheros de configuración creados por EJS cuando se cambien sus opciones de ejecución.
- “export”: Éste es el subdirectorio por defecto que EJS ofrecerá cuando se creen paquetes autoejecutables (ficheros JAR) o ficheros HTML para su distribución.
- “output”: Este directorio es usado por EJS para almacenar ficheros intermedios generados cuando se ejecuta una simulación desde EJS.
- “source”: Éste es el directorio en el que se debe almacenar todos sus ficheros de simulación (ficheros XML), así como ficheros auxiliares de éstas, tales como imágenes GIF o ficheros de datos

Y permite elegir las siguientes opciones:

3.- Cargar el último archivo: si EJS debe cargar al arrancar el fichero de la última simulación usada en la sesión previa.

4.- Minimizar la consola: si se debe minimizar la consola al arrancar.

5.- Idioma: el idioma que debe usar EJS.

6.- Aspecto gráfico: el esquema gráfico para la interfaz gráfica de EJS y la consola. Los esquemas de aspecto gráfico disponibles dependen de su instalación. Tres opciones siempre están disponibles:

- SISTEMA: el aspecto gráfico para ventanas estándar del sistema operativo.

- MULTIPLATFORMA: un aspecto gráfico estándar, algo minimalista, disponible en todas las plataformas.
- POR DEFECTO: el aspecto por defecto que se ha decidido en el fichero de configuración *swing.properties*. (Éste es un fichero opcional que se puede crear en el directorio *lib* de Java.)

Las selecciones de estas opciones se guardan, no se necesita elegirlas de nuevo cada vez que arranca EJS. Simplemente se cambian una vez y los cambios tienen efecto en la próxima sesión.

Finalmente, la pestaña de Opciones Básicas tiene un botón en su parte inferior que permite arrancar la interfaz del usuario.

3.2.1.2 Opciones Avanzadas

La pestaña de Opciones Avanzadas de la consola de EJS (ver figura 11) permite al usuario:

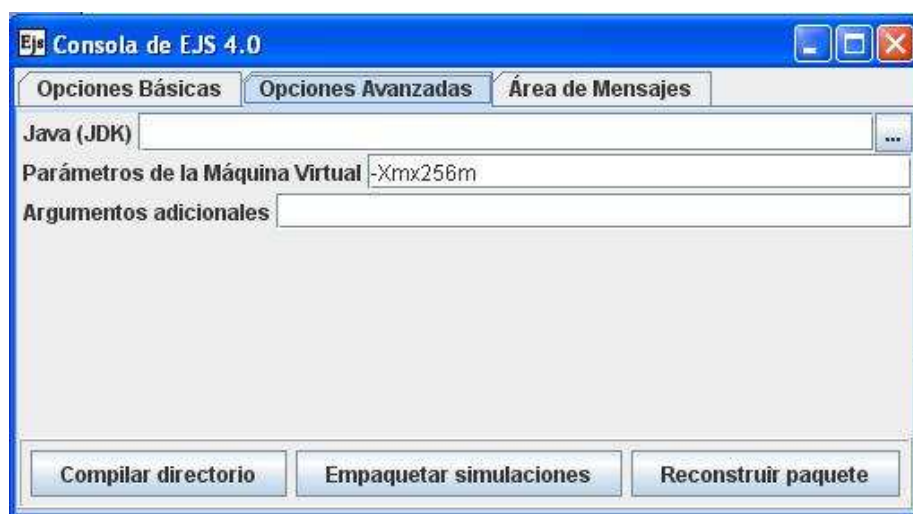


Figura 11: opciones avanzadas



- 1.- Elegir una MV (*Máquina Virtual*) de Java alternativa para ejecutar EJS.
- 2.- Añadir parámetros a la MV Java (opciones de la línea de comando para el propio Java).
- 3.- Especificar argumentos (opciones de la línea de comandos) para EJS.

En la parte inferior de la pestaña de Opciones Avanzadas, hay tres botones que permiten:

1.- **Compilar un directorio:** se abrirá un explorador de ficheros que permite elegir un subdirectorío del directorío “source” de su espacio de trabajo actual. La consola ofrecerá entonces compilar todas las simulaciones (ficheros XML) bajo este subdirectorío. Opcionalmente, se puede pedir a EJS que empaquete cada una de estas simulaciones, una vez compiladas, creando un fichero JAR autoejecutable para cada una.

2.- **Empaquetar simulaciones:** este botón permite crear un único paquete con varias simulaciones (un denominado *Paquete Launcher* – Lanzador) previamente generadas con EJS. Esta funcionalidad también está accesible desde EJS en el menú emergente del botón de empaquetado de la barra de tareas.

3.- **Reconstruir paquete:** permite reconstruir un paquete Launcher creado previamente, con el fin de:

- recompilar las simulaciones del paquete.
- actualizar las librerías del paquete.
- añadir nuevas simulaciones al paquete.

3.2.1.3 Área de Mensajes

La pestaña de Área de Mensajes de la consola de EJS (ver figura 12) muestra algunos mensajes iniciales de arranque, el comando usado para invocar la ventana de edición de EJS, incluidas las opciones de comando, y cualquier otro mensaje de salida que EJS produzca. También dirá si los programas funcionaron correctamente.

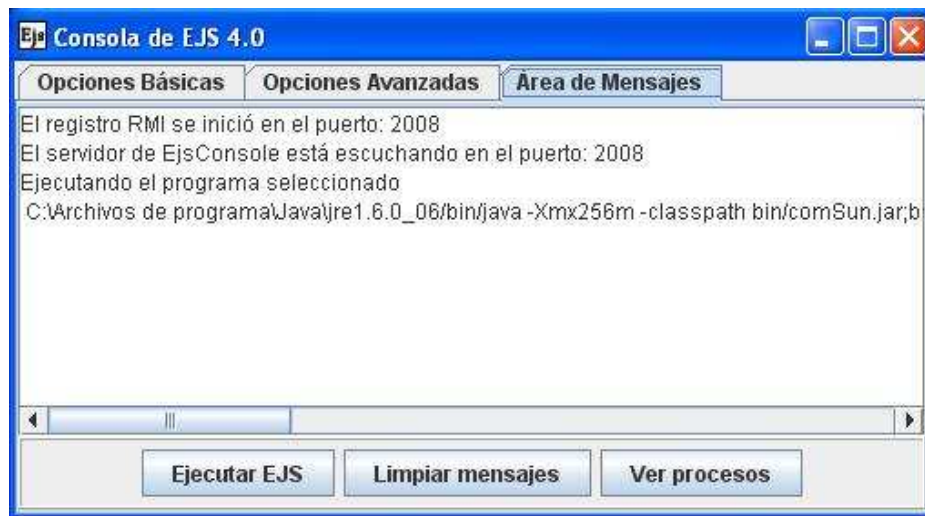


Figura 12: área de mensajes

El área de mensajes también mostrará cualquier mensaje de salida o de error producido por las simulaciones creadas por EJS.

En la parte inferior de la pestaña del Área de Mensajes hay tres botones:

- 1.- Ejecutar EJS: un segundo botón para iniciar la ventana de edición de EJS.
- 2.- Limpiar mensajes: limpia el área de mensajes, lo que resulta útil para borrar mensajes anteriores que pueden confundir.
- 3.- Ver procesos: muestra una ventana de diálogo con una lista de las copias de EJS que están actualmente en ejecución. Estas copias pueden finalizarse (en el caso en

que se colgaran, por ejemplo) haciendo doble-clic en la entrada correspondiente de la lista, o seleccionándola y haciendo clic en el botón “Terminar” de la ventana de diálogo.

3.2.2 Interfaz del usuario

La segunda ventana es la interfaz de usuario de EJS (ver figura 13):

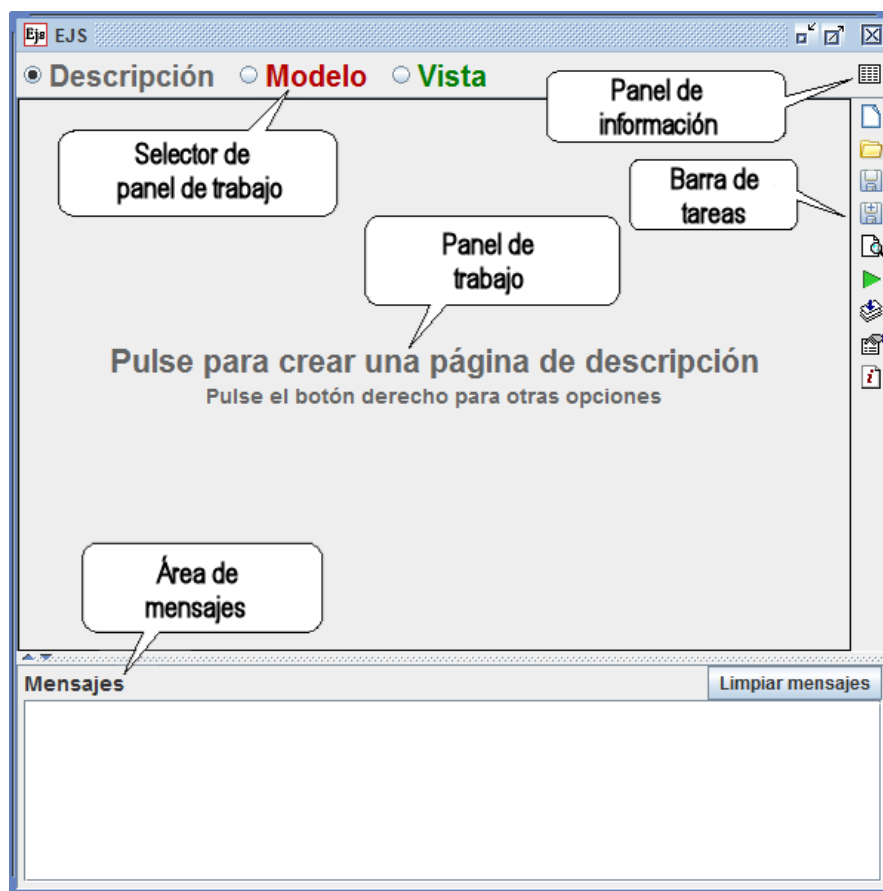


Figura 13: interfaz del usuario. Descripción

La barra de tareas de la derecha proporciona una serie de botones para mostrar un panel de información sobre la simulación, limpiar, abrir y guardar un fichero, buscar en el código, configurar EJS, y mostrar la ayuda de EJS. También proporciona iconos para ejecutar una simulación y para empaquetar una o más simulaciones en un fichero

JAR o exportarlas como applets en páginas web. Haciendo clic con el botón derecho en algunos iconos se puede realizar otras acciones alternativas (pero relacionadas). La parte inferior de la interfaz contiene un área de mensajes donde EJS muestra mensajes de información. La parte central de la interfaz contiene los paneles de trabajo donde se construye la simulación.

EJS proporciona tres paneles de trabajo para construir la simulación, que son: *descripción, modelo y vista*.

3.2.2.1 Descripción

El primer panel, Descripción, permite crear y editar narrativa multimedia basada en código HTML que describe la simulación. Cada página de narrativa aparece en una pestaña de este panel de trabajo y el usuario puede, haciendo clic con el botón derecho, editar la narrativa o importar ficheros HTML escritos con otras herramientas.

Proporciona un lugar donde escribir narrativa introductoria para la simulación. Esta información se mostrará al usuario al arrancar la simulación, cuando se ejecuta como una aplicación, o será parte de las páginas HTML que acompañan a la simulación, cuando se ejecuta como un applet.

3.2.2.2 Modelo

El segundo panel de trabajo, Modelo, está dedicado al proceso de modelado. Este panel se usa para crear variables que describen el modelo de nuestra simulación, para inicializar estas variables, y para escribir algoritmos que describen cómo varían estas variables con el tiempo.

Dentro del panel Modelo podemos encontrar cinco subpaneles: *Variables*, *Inicialización*, *Evolución*, *Relaciones fijas* y *Propio*. Al pulsar sobre el interior del subpanel creamos una nueva página a la que pide que se de un nombre. Se crea la página mostrando en la parte superior de la misma una pestaña. Haciendo clic con el botón derecho sobre la misma el menú desplegable nos permitirá entre otras opciones crear nuevas páginas de forma que nuestras variables puedan ser agrupadas de forma más clara. De la misma forma en ese menú desplegable puede elegir activar/desactivar la página cuando deje de utilizar esa página en su modelo sin necesidad de eliminarla.

3.2.2.2.1 Variables

Se llama variables a todas aquellas magnitudes que intervienen en el modelo, tanto si se modifican a lo largo del tiempo de ejecución como si permanecen constantes. Éstas pueden verse modificadas por la dinámica interna de la simulación aunque también por parte del usuario. En ocasiones la modificación de una variable produce una modificación en otras variables, entonces decimos que existen *ligaduras*.

Para crear una nueva variable lo primero es darle un nombre, especificar un tipo y tratándose de vectores o matrices su dimensión.

1.- Tipos de variables:

Dependiendo del uso que se vaya a hacer de ellas el ordenador necesita que se especifique de qué tipo es cada una de nuestras variables. Aunque en Java existen algunos tipos más, destinados a optimizar el uso de la memoria, EJS funciona con los siguientes tipos:

Boolean: para valores de verdadero o falso.



Int: para valores enteros.

Double: para números en coma flotante, es decir, números realesl.

String: para caracteres y textos.

Object: tipo que nos ofrece la programación orientada a objetos de Java y que permite a programadores más avanzados crear sus propios tipos.

2.- Nombrar variables:

Tanto para dar nombre a las variables como al resto de los elementos de la simulación tales como paneles, métodos propios del modelo y elementos de la vista hay que tener en cuenta algunas restricciones y consejos.

En primer lugar hay un grupo de palabras que no pueden usarse ya que son palabras reservadas por Java o por el propio EJS: *boolean, break, byte, case, catch, char, continue, default, do, double, else, float, for, getSimulation, getView, if, initialize, instanceof, int, long, Object, reset, return, short, step, String, switch, synchronized, throws, try, update, while*.

Es aconsejable además, para evitar conflictos entre los distintos nombres y confusiones con el código propio de EJS que:

- Los nombres de cada variable sean únicos en cada modelo.
- Estén formados únicamente por caracteres alfanuméricos y el carácter “_”, empezando siempre por un carácter alfabético.
- Aunque no sea estrictamente obligatorio, que el primer carácter en los nombres de variables y métodos y los elementos de vista sea minúscula.

- Utilizar nombres descriptivos. En caso de que sean necesarias más de una palabra no se debe utilizar espacios en blanco si no escribir algo así como *centroMasas*.
- No se debe utilizar nombres que comiencen por “_” ya que EJS puede hacer uso de éstos nombres.

3.- Valor inicial:

El valor inicial de una variable puede ser una constante o una expresión sencilla que establezcamos en la casilla destinada para ello. Si esta casilla se deja en blanco EJS le da un valor por defecto (el valor 0 si se trata de variables de tipo numérico). Se puede además establecer ese valor inicial más tarde en el panel de inicialización o mediante una ecuación de ligadura.

3.2.2.2 Inicialización

Si bien en la creación de las variables se pueden inicializar con un valor constante o una expresión sencilla dentro de la casilla *Valor Inicial* en algunos casos se necesita una inicialización más compleja, como por ejemplo si se necesita hacer unos cálculos previos. Las páginas que se pueden crear en este subpanel son el lugar donde escribir el código Java necesario. En la parte inferior del panel se puede añadir un comentario sobre el código para facilitar su comprensión.

Las páginas se ejecutarán por orden de izquierda a derecha aunque puede, haciendo clic con el botón derecho sobre la pestaña correspondiente, reordenarlas (*Mover esta página a la derecha*).

3.2.2.2.3 *Evolución*

El sistema se desarrollará de forma autónoma desde el estado actual de las variables, a un nuevo estado, simulando así el paso del tiempo. A las reglas de transición que nos llevan del estado actual al nuevo estado las llamamos *ecuaciones de evolución*.

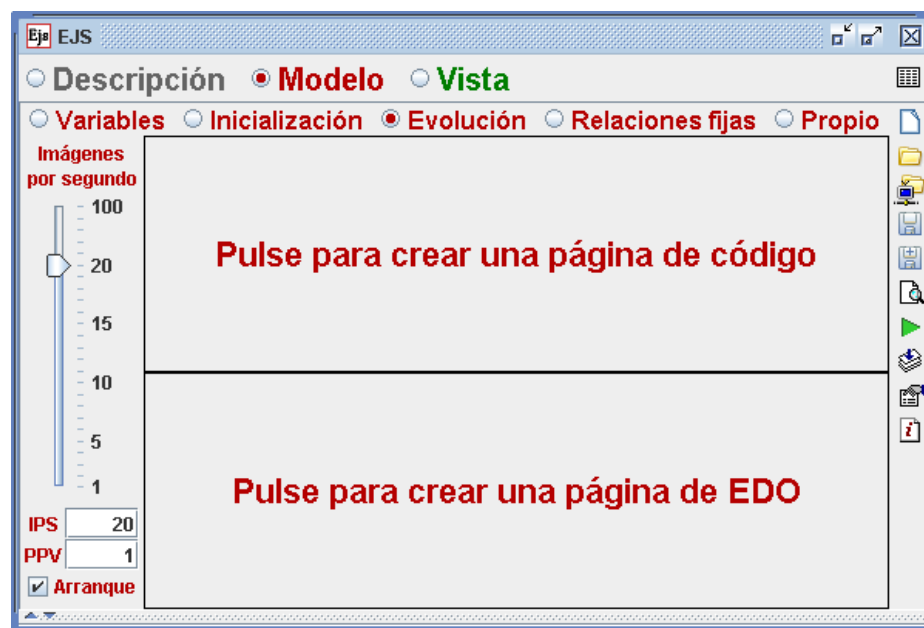


Figura 14: página de evolución

Algunas veces estas ecuaciones tendrán una formulación matemática directa. En estos casos se habla de sistemas discretos. En otras situaciones, debido a la discretización de modelos continuos, será necesario utilizar ecuaciones diferenciales. Por esta razón este tercer subpanel está dividido en dos partes. En la parte superior se puede crear una página para editar código Java. Éste es el indicado para simulaciones de modelos discretos. Sin embargo para aquellos modelos en los que se necesita la resolución numérica de ecuaciones diferenciales ordinarias, EJS dispone, en la mitad inferior, de un editor de EDO que le facilitará la tarea de programar el código Java necesario.

Llegado el momento de trabajar en la evolución hay que elegir entre uno

de los dos editores, el de código Java o el especializado en EDO. Si el modelo requiere de los dos tipos, el menú desplegable para estas páginas le permite crear nuevas páginas de ambos tipos.

Además, en la parte izquierda del subpanel hay un deslizador que permite modificar el número de *imágenes por segundo (IPS)*. Este valor se ve reflejado en el campo numérico justo debajo y puede ser modificado directamente desde ahí. Con el *IPS* decidimos cuántas veces queremos que se pinte nuestra simulación en cada segundo. El otro campo numérico, que se muestra bajo el nombre *PPV, pasos por visualización*, especifica cuántas veces queremos que avance el modelo antes de volver a pintar. Con la casilla *Arranque* se indica si deseamos que la simulación empiece al comenzar el programa.

3.2.2.2.4 Relaciones fijas

En ocasiones la modificación de una variable produce una modificación en otras variables, entonces se dice que existen *relaciones fijas*.

Un criterio útil (aunque quizá no siempre válido) para distinguir estos dos tipos de relaciones es examinar la expresión que se usa para calcular el valor de una variable en un momento dado y si este valor depende del valor actual de esa misma variable, en tal caso es probable que se trate de una ecuación de evolución. Por el contrario, si el nuevo valor de la variable se calcula sólo a partir de los valores de otras variables será una relación fija.

La razón por la que no se deben incluir estas relaciones entre las ecuaciones de evolución es la existencia de una segunda fuente de cambios: la interacción directa del usuario con la simulación. Si las relaciones fijas han de ser válidas siempre, entonces deberían mantenerse cuando el usuario cambia

alguna de estas variables, incluso si la simulación está parada. Por esto, es conveniente identificar claramente y escribir de forma separada estos dos tipos.

Éste cuarto subpanel funciona de la misma forma que el de *Evolución*. Aquí deben incluirse las ecuaciones de relaciones fijas de su modelo transcritas a lenguaje Java.

3.2.2.2.5 Código Propio

Este último subpanel del *Modelo* es el indicado para definir sus propios métodos en Java (también llamadas funciones o subrutinas).

Este subpanel funciona de la misma forma que los dos anteriores, en cuanto a la creación y uso de páginas, aunque existen dos diferencias importantes.

Por un lado está el uso que EJS hará del código que se escriba en estas páginas. A diferencia del código del resto de editores el cual pasa a formar parte de la simulación directamente, estos métodos no se utilizarán a menos que se invoquen desde otro lugar de la simulación.

Por otra parte en este panel tiene total libertad para escribir su código Java, el único requisito es que sean métodos Java válidos.

Crear un método Java requiere, antes de empezar a escribir el código necesario, que se le dé un nombre, que se indique el tipo del valor de salida (si es que hay un valor de salida) y que se especifique la *accesibilidad* del método. Además hay que decidir si acepta parámetros de entrada y de qué tipo.

- El **nombre** para el método puede ser cualquiera pero teniendo en cuenta las convenciones que se explicaron en la creación de variables.

- El término ***accesibilidad*** se refiere a qué partes de la simulación podrán hacer uso de este método. Si el método es declarado *public* entonces será universalmente visible y puede ser usado desde cualquier parte de la simulación e incluso desde fuera de la simulación utilizando *JavaScript*. Si el método no es declarado público (ya sea utilizando la palabra *private* o simplemente nada) entonces el método sólo podrá ser utilizado desde el modelo.

- El **tipo de valor de salida** es el tipo de aquello que devuelva la función. Si la función no devuelve ningún valor se indicará con el tipo especial *void*. Si devuelve un valor habrá que decir de qué tipo Java es, ya sea simple o compuesto. En tal caso el código del método tendrá que terminar con una sentencia *return* que indique el valor devuelto.

- Los **parámetros** se indican con una lista separada por comas de parejas tipo-nombre que declaren variables locales que podrán utilizarse a lo largo del método. Esta lista se escribe entre paréntesis justo después del nombre del método. Si el método no necesita parámetros de entrada la lista estará vacía.

3.2.2.3 Vista

El tercer panel de trabajo, Vista, está dedicado a la tarea de construir la interfaz gráfica de usuario de la simulación, que permitirá a los usuarios finales controlar la simulación y mostrar sus salidas. La interfaz se construye seleccionando elementos de unas paletas (ver figura 15) y añadiéndolos al denominado *Árbol de elementos* de la vista. Por ejemplo, la paleta *Interfaz* contiene botones, deslizadores y campos de entrada, y la paleta *Elementos de dibujo 2D* contiene elementos para dibujar datos 2D.

Para crear una simulación hay que reunir los componentes como las piezas de un rompecabezas con los elementos del panel de la vista. Los elementos de la vista están organizados en los siguientes tres grupos:

- [Elementos de la Interfaz](#)
- [Elementos de Dibujo 2D](#)
- [Elementos de Dibujo 3D](#)

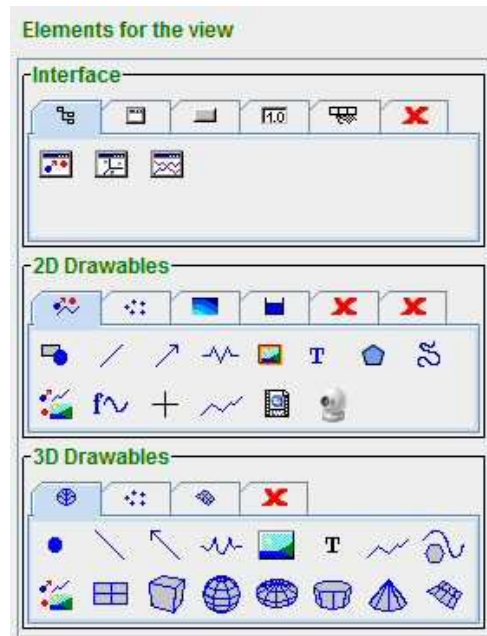


Figura 15: elementos de la vista

Los elementos de la interfaz: son utilizados para construir la interfaz. Éstos elementos están basados en elementos estándar de Java para interfaces, tales como ventanas paneles y botones.

Los elementos de Dibujo 2D: son elementos de dibujos para paneles bidimensionales. Por tanto estos elementos solo pueden encontrarse dentro de paneles de dibujo o con ejes 2D.

Los elementos de Dibujo 3D: son elementos de dibujos para paneles tridimensionales. Por tanto estos elementos solo pueden encontrarse dentro de paneles de dibujo 3D.



Para conectar una variable del modelo con el elemento de vista apropiado, simplemente se escribe el nombre de la variable en cuestión en el campo de texto que se encuentra en el editor de propiedades del elemento.

Esta conexión es bidireccional, lo que significa que si la variable del modelo cambia, el elemento de la vista recibe el nuevo valor de forma automática y cambia consecuentemente. Y al contrario, si el usuario interactúa con el elemento de forma que sus propiedades se modifican de alguna forma, el nuevo valor pasa a la variable del modelo.



Capítulo 4

Depósito

4.1 Enunciado del problema

El sistema de la figura representa un depósito de área constante A , alimentado por un caudal de entrada $q_e(t)$. El depósito pierde líquido con un caudal de salida $q_s(t)$.

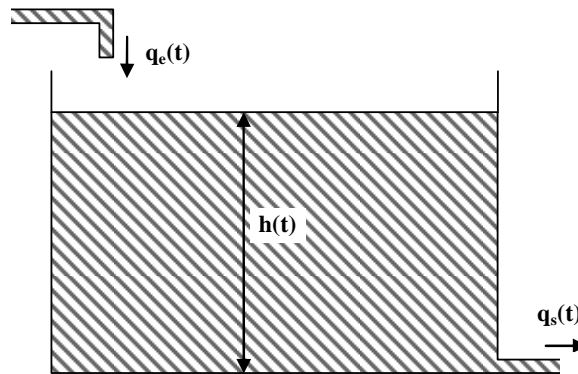


Figura 16: depósito

Se pretende encontrar un modelo lineal que permita conocer la altura del líquido $h(t)$, cuando el caudal de entrada $q_e(t)$ sufra pequeñas variaciones respecto del punto de funcionamiento y comparar su evolución con la evolución del modelo real.

Las ecuaciones físicas que determinan el comportamiento del sistema son:

$$q_e(t) - q_s(t) = A \frac{d h(t)}{dt} \quad (1)$$

$$q_s(t) = k \sqrt{h(t)} \quad (2)$$

La ecuación (1) representa el incremento de volumen del líquido contenido en el depósito, mientras que la ecuación (2) por su parte corresponde al principio de Torricelli.

Este modelo no lineal nos permite determinar el punto de equilibrio, (aquel en el que la altura h permanece constante):

$$q_{e0} = q_{s0}$$

y por lo tanto:

$$h_0 = \left(\frac{q_{s0}}{k} \right)^2$$

De modo que para cada caudal de entrada constante q_{e0} , hay una altura de equilibrio h_0 .

Si estando trabajando en estas condiciones estabilizadas el caudal de entrada experimenta alguna variación, la altura del depósito experimentará también una variación temporal que, si el sistema es estable, acabará en un nuevo punto de equilibrio diferente del anterior.

4.2 Resolución del problema

Conociendo el enunciado del problema se procede a dar una solución mediante la herramienta software EJS.

Hay que aclarar que se simularán dos modelos de depósitos, un modelo no linealizado y otro linealizado. Se parte del modelo no linealizado, y con éste se obtiene el modelo linealizado para poder comparar los resultados.

4.2.1 Modelo no lineal

Hay que estudiar el modelo por partes para poder definirlo dentro del programa, estas son: variables, ecuaciones y vista.

4.2.1.1 Variables

Para comenzar se toma el modelo no lineal, lo primero es realizar la lista de variables dentro del panel modelo, las variables que se van a utilizar son:

- q_e : caudal de entrada
- q_s : caudal de salida
- h : altura del líquido en el depósito
- A : área del depósito
- k : constante
- t : tiempo
- dt : diferencial de tiempo

- El caudal de entrada es variable, pero se da el valor por defecto de $1\text{ m}^3/\text{s}$, ya que en los ejemplos que se van a realizar, se parte de un caudal de entrada de $1\text{ m}^3/\text{s}$ y se aumenta a $1,1\text{ m}^3/\text{s}$ para ver el comportamiento del sistema. Pero se podría poner cualquier valor.

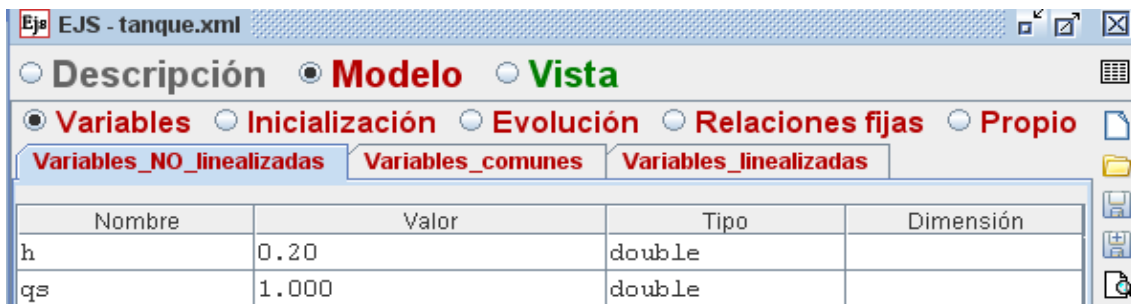
- El caudal de salida se toma de valor igual al caudal de entrada para el comienzo: $1\text{ m}^3/\text{s}$. Aunque en cuanto arranque la simulación este caudal cambiará ya que el caudal de salida depende de la altura (principio de Torricelli) y la altura cambiara para ponerse en equilibrio.

- A la altura se le da el valor por defecto de $0,2\text{ m}$ para ver como se estabiliza el sistema y se queda en equilibrio, partiendo de una altura que no es la de equilibrio; que

para el caso de $q_e = q_s = 1 \text{ m}^3/\text{s}$ sería $h = 0,25\text{m}$, por lo que se le puede dar cualquier valor por defecto, el único interés de darle un valor distinto a $0,25\text{m}$ es ver como se estabiliza el sistema y va al punto de equilibrio.

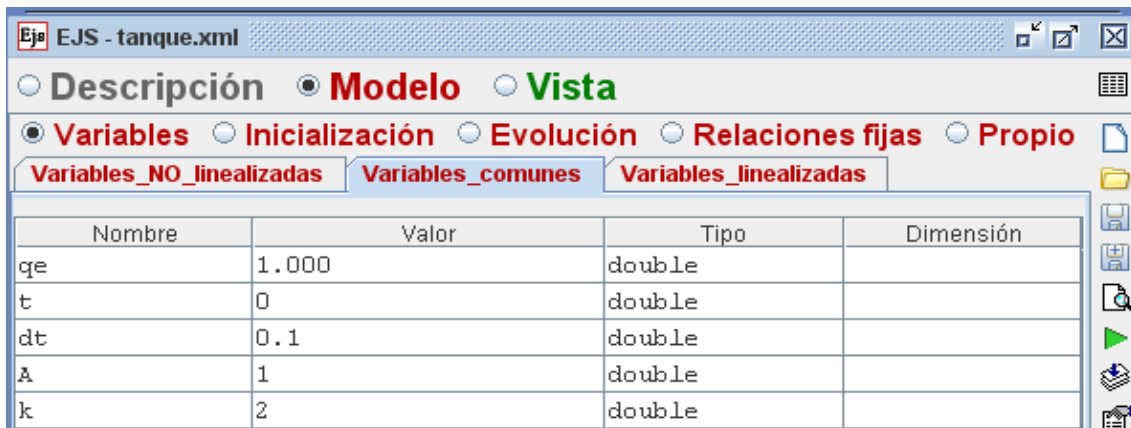
- El área del depósito para este ejercicio tiene un valor de 1m^2 .
- La constante del principio de Torricelli se le da un valor de $k = 2$.
- Se define la variable tiempo para las ecuaciones diferenciales y para ver como evoluciona el sistema en el tiempo. Se le da un valor de $t = 0$ para que comience en el tiempo 0s de la gráfica.
- El diferencial de tiempo, ha de tomar un valor pequeño, así que se le da el valor $dt = 0,1\text{s}$ pero podría tomar cualquier otro también pequeño. Cuanto más pequeño sea el diferencial de tiempo más lento será el proceso de simulación de estabilización del sistema.

Hay que añadir que algunas de estas variables son exclusivas del modelo lineal y otras que son comunes con el modelo no lineal, por lo que se procede a agruparlas en 3 paginas de variables, las variables del modelo no lineal (ver figura 17), las comunes (ver figura 18) y las del modelo lineal.



| Nombre | Valor | Tipo | Dimensión |
|--------|-------|--------|-----------|
| h | 0.20 | double | |
| qs | 1.000 | double | |

Figura 17: variables no linealizadas



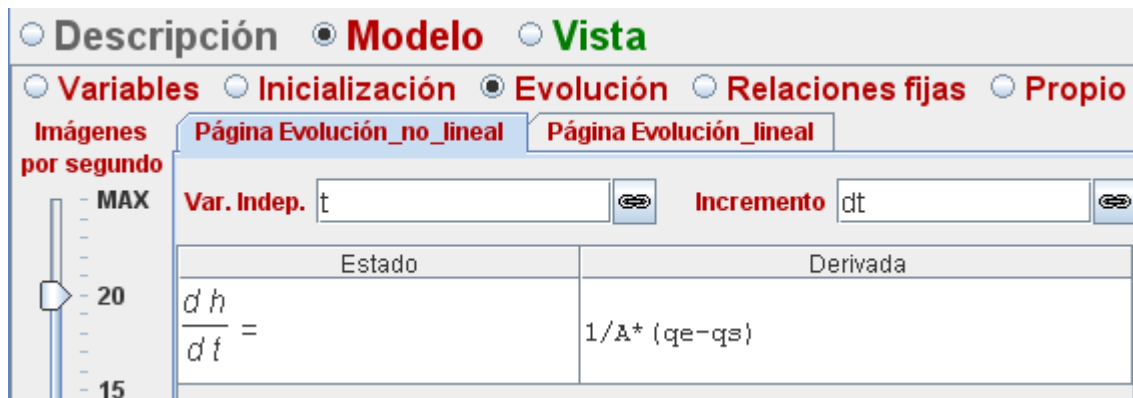
| Nombre | Valor | Tipo | Dimensión |
|--------|-------|--------|-----------|
| qe | 1.000 | double | |
| t | 0 | double | |
| dt | 0.1 | double | |
| A | 1 | double | |
| k | 2 | double | |

Figura 18: variables comunes

4.2.1.2 Ecuaciones

Una vez definidas las variables, dentro del panel modelo, se pasa a escribir las ecuaciones necesarias. Estas ecuaciones están definidas en el enunciado.

La ecuación $q_e(t) - q_s(t) = A \frac{dh(t)}{dt}$ se coloca en la pestaña de evolución ya que es una ecuación diferencial.



| Estado | Derivada |
|-------------------|---------------------|
| $\frac{dh}{dt} =$ | $1/A * (q_e - q_s)$ |

Figura 19: ecuación evolución

Sin embargo la ecuación $q_s(t) = k\sqrt{h(t)}$ se coloca en la pestaña de relaciones fijas, ya que el principio de Torricelli se cumple siempre y no depende de la variable tiempo, sino de la altura.

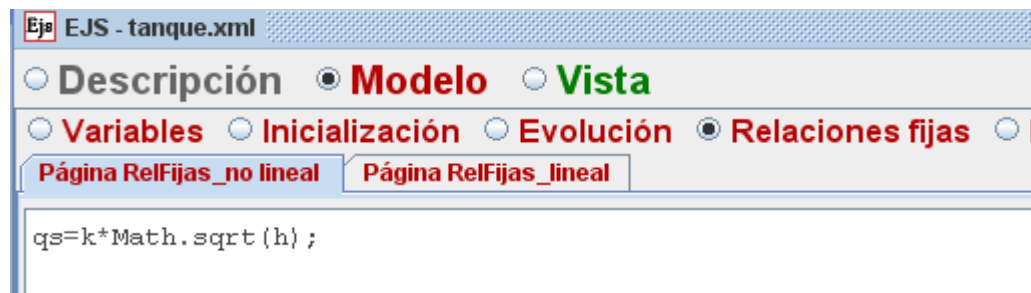


Figura 20: ecuación relación fija

Nota: a la hora de escribir las ecuaciones hay que dejar la variable a calcular despejada en el primer miembro de la ecuación para que el programa no de errores de compilación.

4.2.1.3 Vista

Una vez definidas las variables y las ecuaciones, hay que asociar esas variables a unas figuras que simulen esas variaciones de altura, esto se hace desde el panel Vista.

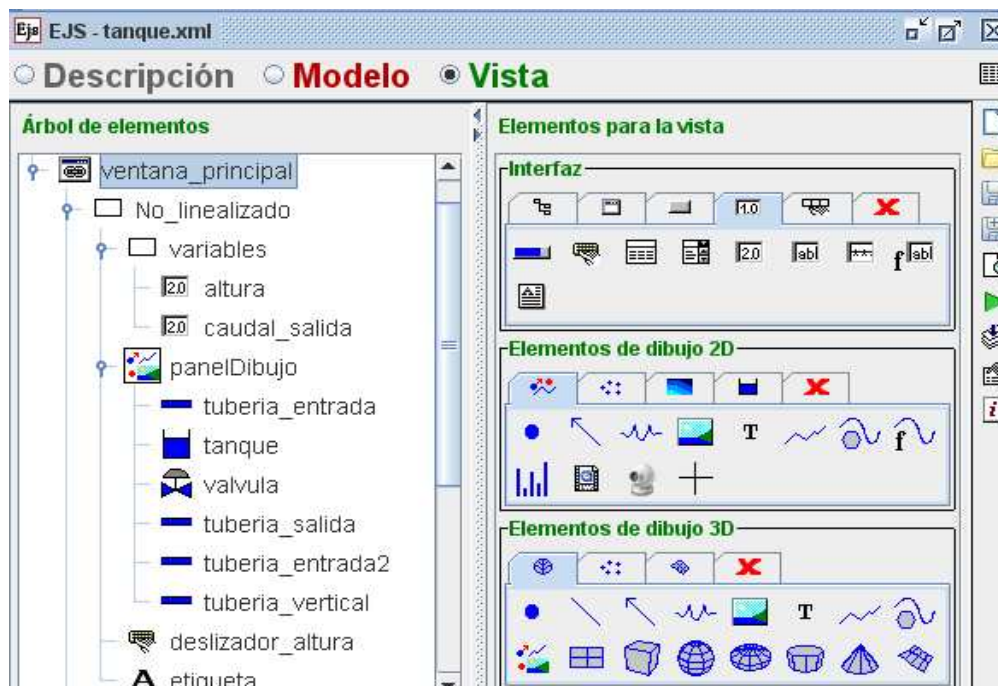


Figura 21: elementos de la vista

Lo primero es colocar la ventana principal, donde se colocará la gráfica de la evolución de la altura con el tiempo y el dibujo del depósito con las casillas de las variables con los valores que van tomando. Cada dibujo o gráfica va colocado en un panel contenedor básico.

Dentro del contenedor que recibe el nombre de “No linealizado” se coloca:

- Un contenedor con las variables que pueden interesar como el caudal de salida y la altura.
- Un panel de dibujo donde se colocan los elementos necesarios para simular el contenedor y las tuberías.
- Un deslizador para que el usuario pueda cambiar la altura y ver como sin variar el caudal de entrada, la altura vuelve a su posición en la que se encontraba, que es la de equilibrio.
- Una etiqueta para ponerle un título a la simulación, en este caso, diferenciar la simulación no linealizada de la linealizada.

Una vez colocados todos los elementos visuales se asocian las variables a los elementos:

Se puede observar en la figura 22 un ejemplo de cómo modificando las propiedades del depósito se asocia el nivel con la altura (h), lo cual va a simular como sube y baja en nivel del agua.

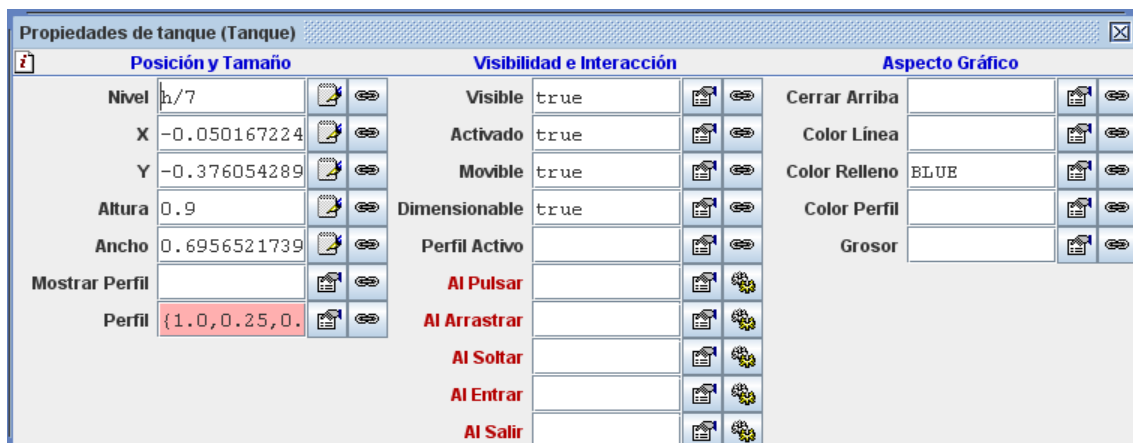


Figura 22: propiedades de la altura del agua

En este caso se divide la altura entre 7 para que el nivel del líquido no se salga de la ventana del dibujo, es decir que la altura aumente de forma más lenta.

Dentro de la ventana de propiedades se modifican los atributos del dibujo para acercarlo más al ejercicio en particular. En este caso se activa la opción dimensionable y movable para poder darle un tamaño adecuado desde la vista de la simulación.

Una vez colocado y dimensionado el depósito hay que incluir la gráfica, para ello dentro de la ventana principal se coloca un panel contenedor básico, al que se le llama gráfica y dentro de él se incluye:

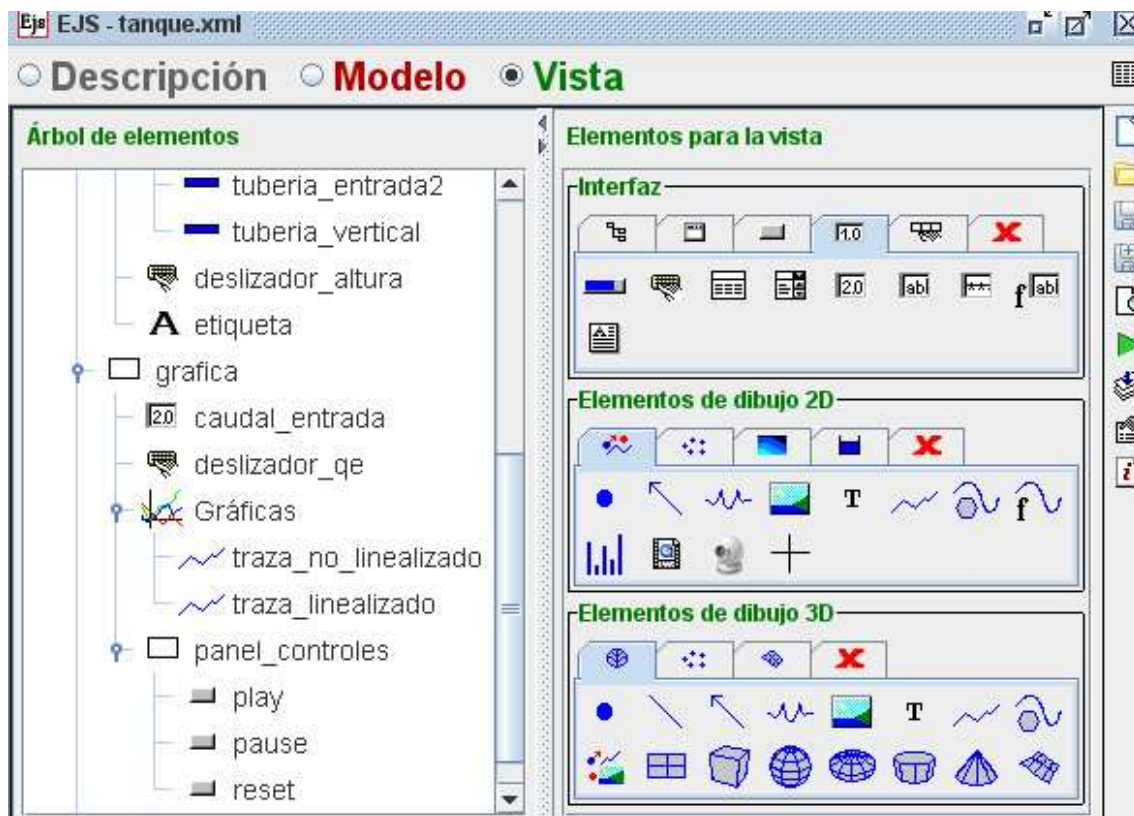


Figura 23: elementos de la gráfica

- Una casilla para informar numéricamente del caudal de entrada.
- Un deslizador para que el usuario pueda modificar el caudal de entrada deslizando el ratón sobre la barra.
- Un panel de gráficas donde se incluyen tantas trazas como se necesiten (en este caso dos).
- Un panel de controles para controlar la simulación, con tres botones: play, pause y reset. Estos botones cumplen su función indicándoselo en el campo acción y el campo activado para indicar cuando se encuentra activo dicho botón (ver figuras 24, 25 y 26).



Figura 24: propiedades del botón play



Figura 25: propiedades del botón pause



Figura 26: propiedades del botón reset

Ya se encontraría listo el modelo para ser simulado, pero lo que se pretende con este ejercicio es ver la comparación del sistema linealizado con el sistema no linealizado.

4.2.2 Modelo lineal

A partir del modelo no lineal se encuentra el punto donde la altura permanece constante, ese es el punto de equilibrio entorno al cual hay que linealizar.

4.2.2.1 Variables

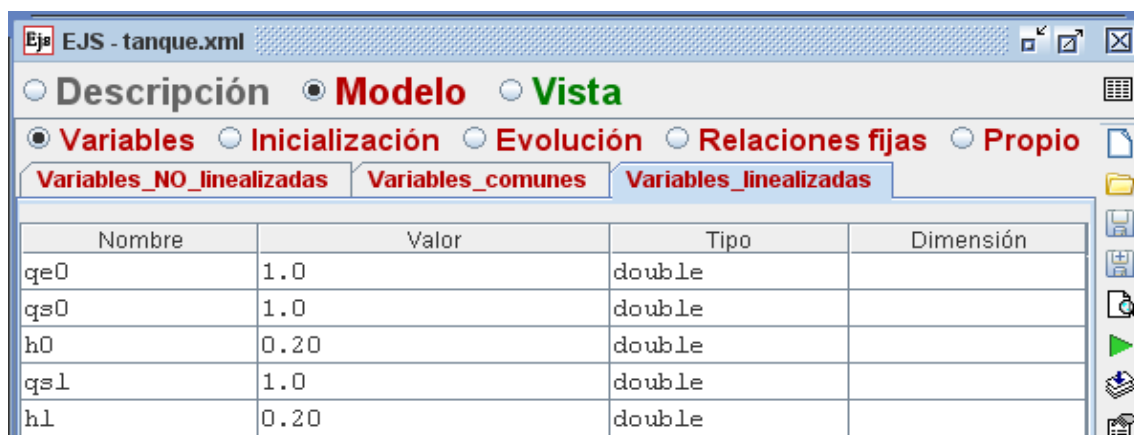
Hay que definir variables linealizadas y las variables que definen el punto de equilibrio.

Las variables linealizadas son:

- q_{sl} : caudal de salida linealizado
- h_l : altura linealizada

Las variables que definen el punto de equilibrio son:

- q_{e0} : caudal de entrada de equilibrio
- q_{s0} : caudal de salida de equilibrio
- h_0 : altura de equilibrio



| Nombre | Valor | Tipo | Dimensión |
|----------|-------|--------|-----------|
| q_{e0} | 1.0 | double | |
| q_{s0} | 1.0 | double | |
| h_0 | 0.20 | double | |
| q_{sl} | 1.0 | double | |
| h_l | 0.20 | double | |

Figura 27: variables linealizadas

El resto de variables (q_e , k , A , t , dt) están definidas en el modelo no lineal, las cuales se pueden utilizar también para este modelo, son las llamadas variables comunes.

4.2.2.2 Ecuaciones

A partir del modelo no lineal se determina el punto de equilibrio, que será aquel en el cual la altura permanece constante, es decir, la variación de altura será nula. Expresado matemáticamente:

$$h(t) = h_0 \rightarrow \frac{dh(t)}{dt} = 0$$

Y sustituyendo estos valores en la ecuación (1), se obtiene que el sistema estará en equilibrio cuando los caudales de entrada y salida se igualen:

$$q_{e0} = q_{s0}$$

Para un caudal de entrada constante q_{e0} , de la ecuación (2): $q_{e0} = k \cdot \sqrt{h(t)}$; con lo que la altura de equilibrio h_0 , se calcula según la expresión:

$$h_0 = \left(\frac{q_{e0}}{k} \right)^2 \quad (3)$$

A continuación se linealiza el sistema en el punto de equilibrio, partiendo de las ecuaciones (1) y (2), obteniéndose las siguientes ecuaciones:

$$\Delta q_e(t) - \Delta q_s(t) = A \cdot \dot{\Delta h(t)} \quad (4)$$

$$\Delta q_e(t) = k \cdot \frac{1}{2 \cdot \sqrt{h_0}} \cdot \Delta h(t) \quad (5)$$

En el sistema linealizado las variables son incrementales, referidas al punto de equilibrio. Así, cuando se produzca una variación en el caudal de entrada y este pase de q_{e0} a q_{s0} , el sistema evolucionará hasta que se estabilice en un nuevo punto de equilibrio, que para el sistema real vendrá dado, según la ecuación (3), por:

$$h_1 = \left(\frac{q_{e1}}{k} \right)^2$$

Sin embargo, en el sistema linealizado las variables son incrementales, con lo que se tendrá: $\Delta q_{e01} = q_{e1} - q_{e0}$. En el nuevo punto de equilibrio no habrá variaciones de altura, por lo que, según (4) $\Delta q_e = \Delta q_s$. Teniendo esto en cuenta, a partir de (5) se obtiene el nuevo punto de equilibrio para el sistema linealizado:

$$\Delta h_{01} = \frac{2 \cdot \sqrt{h_0}}{k} \cdot \Delta q_{e01} \quad (6)$$

Con esta expresión se calculan las variaciones de altura con relación al punto de funcionamiento. Para calcular la altura de líquido en el depósito hay que sumar la altura en el punto de equilibrio h_0 , con lo que se obtiene:

$$h_1 = h_0 + \Delta h_{01} \quad (7)$$

De donde tenemos que $h_1 - h_0 = \Delta h_{01}$; en general, cualquier incremento se puede poner en forma de resta del punto final menos inicial. Como se observa al escribir la ecuación (4) (ver figura 28).

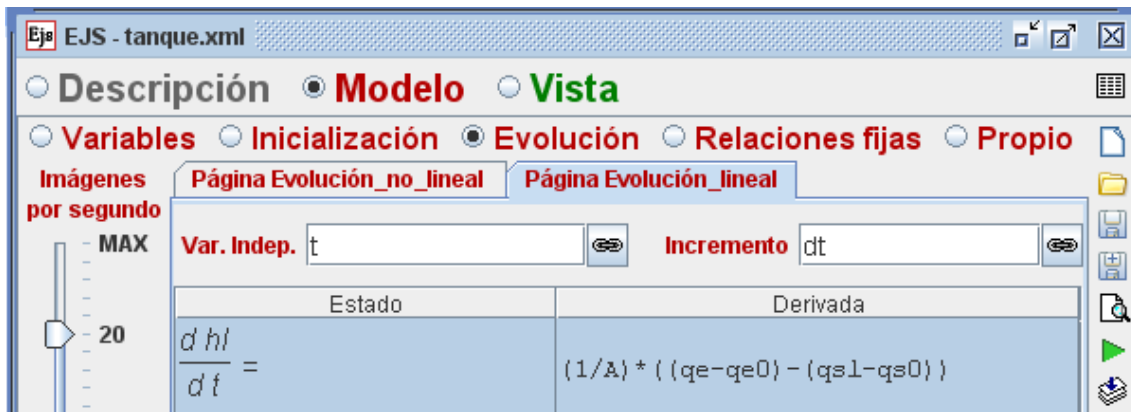


Figura 28: ecuación diferencial

La ecuación (5) y las ecuaciones que definen el punto de equilibrio se escriben la pestaña de relaciones fijas (ver figura 29).

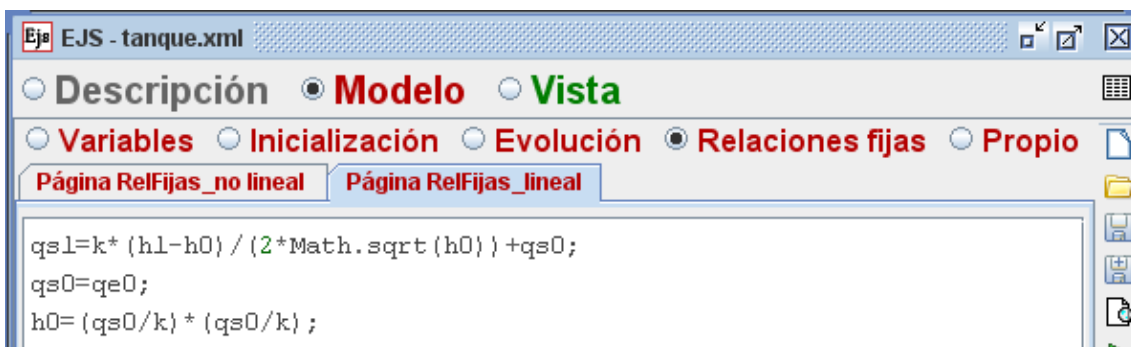


Figura 29: ecuaciones relaciones fijas

4.2.2.3 Vista

El desarrollo del depósito linealizado es el mismo proceso que el del no linealizado, la única diferencia consiste en el momento de enlazar las variables con los dibujos, que en este caso hay que unirlo con las variables linealizadas (ver figura 30).

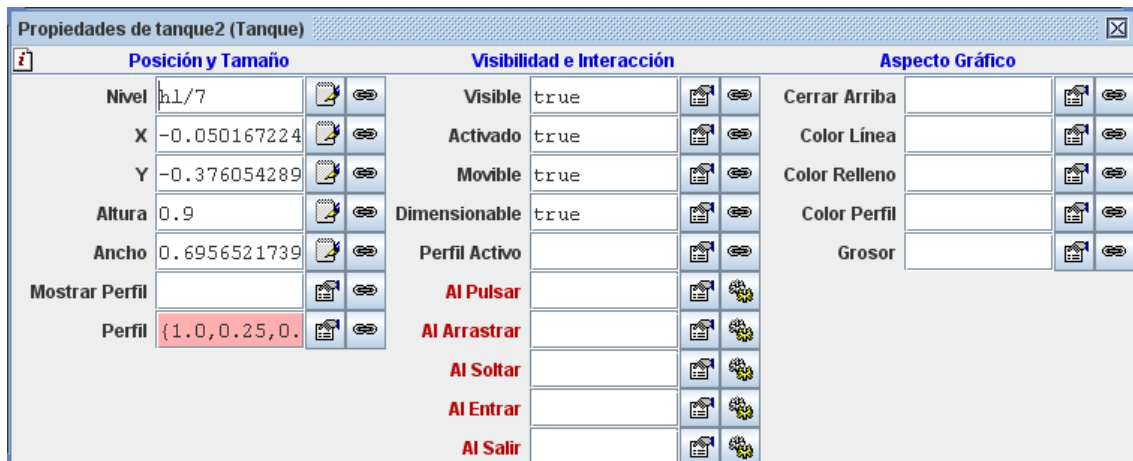


Figura 30: propiedades de la altura del agua

También se incluyen tres casillas donde se ve en torno a qué punto de equilibrio se linealiza (h_0 , q_{e0} , q_{s0}), pudiendo ser modificado este para ver los resultados que se obtienen linealizando en distintos puntos de equilibrio.

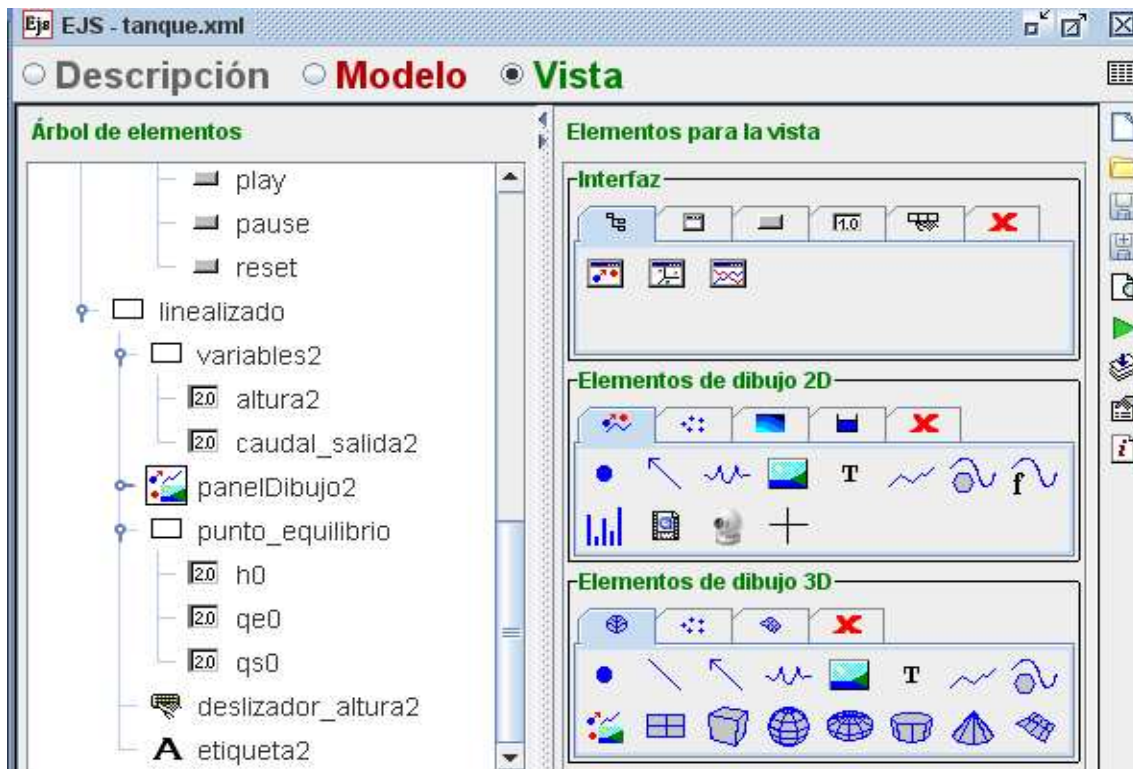


Figura 31: elementos de la vista

Una vez realizado este trabajo solo queda ver la simulación:

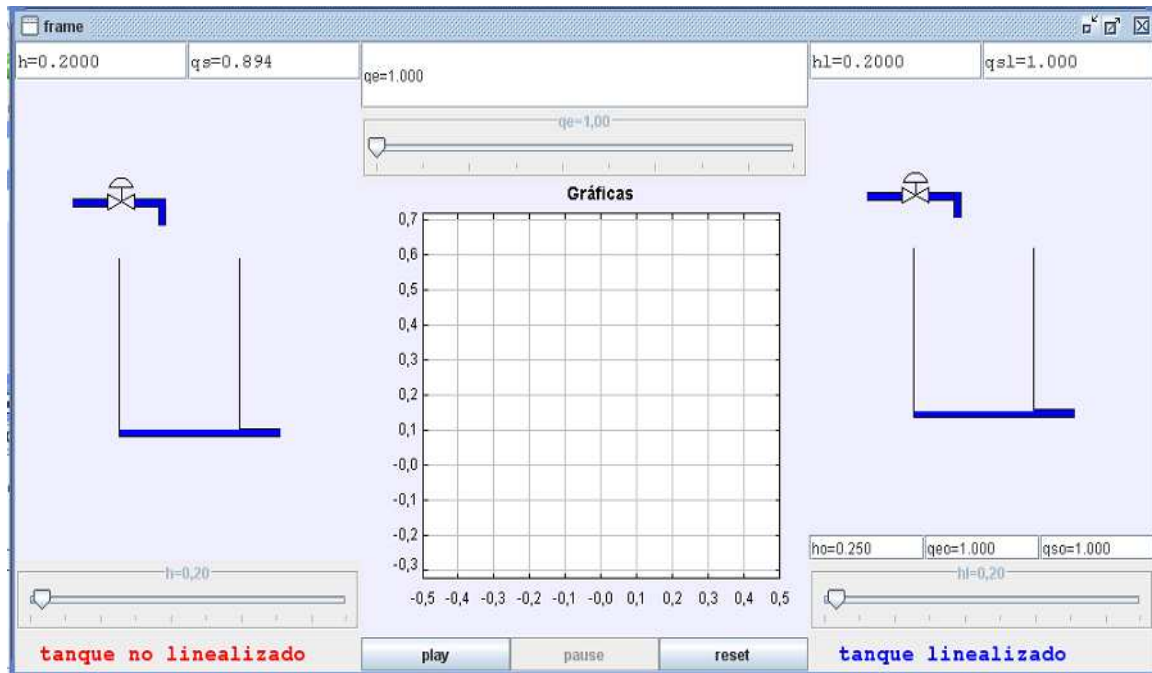


Figura 32: simulación final



Capítulo 5

Barco

5.1 Enunciado del problema

Se tiene un barco de sección triangular (triángulo equilátero) y longitud L . Cuando el barco se introduce en el agua se sumerge una distancia $x(t)$.

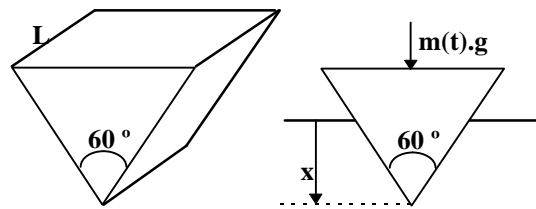


Figura 33: esquema del barco

- Considerar que la fuerza de rozamiento del agua sobre el barco es igual a $k \frac{dx(t)}{dt}$ ($k = 1600 \text{ N}\cdot\text{s/m}$).
- La fuerza estática que ejerce el agua sobre el barco es igual al peso del volumen de líquido desalojado (principio de Pascal).
- No considerar los movimientos laterales.
- La densidad del agua es $\rho = 1000 \text{ kg/m}^3$, la longitud del barco es $L = 4\text{m}$ y su masa sin carga $m_0 = 204 \text{ Kg}$.

Se pretende calcular que altura deberá tener el barco para que no le entre agua en ningún momento si de pronto se sube a la misma una persona de 100 Kg , comparando la respuesta del sistema lineal con la del sistema real.

El comportamiento del sistema puede modelarse según la ecuación diferencial:

$$m(t)g - k \dot{x}(t) - (\rho g L / \sqrt{3}) x(t)^2 = m(t) \ddot{x}(t)$$

5.2 Resolución del problema

Conociendo el enunciado del problema se procede a dar una solución mediante la herramienta software EJS.

Hay que aclarar que se simularán dos modelos de barcos, un modelo no linealizado y otro linealizado. Se parte del modelo no linealizado, y con éste se obtiene el modelo linealizado para poder comparar los resultados.

5.2.1 Modelo no lineal

Hay que estudiar el modelo por partes para poder definirlo dentro del programa, estas son: variables, ecuaciones y vista.

5.2.1.1 Variables

Para comenzar se toma el modelo no lineal, lo primero es realizar la lista de variables dentro del panel modelo, las variables que se van a utilizar son:

- x: profundidad
- v: velocidad
- a: aceleración
- m: masa
- t: tiempo
- dt: diferencial de tiempo
- g: gravedad



- k: constante
- ρ : densidad del agua
- l: longitud de la base del barco

- A la profundidad se le da el valor de 0,2 m para que se vea el barco que está dentro del agua, aunque en cuanto se simula el ejercicio toma otro valor distinto ya que calcula la profundidad de forma instantánea.

- La velocidad y la aceleración toman un valor igual a 0, ya que se estudia el caso cuando el barco está en la posición de equilibrio según su masa. Aunque estas dos variables pueden ser modificadas en la simulación cambiando su valor pinchando sobre el recuadro donde se ve su valor.

- La masa se puede modificar desde el recuadro donde se observa su valor o con el deslizador, y así se ve como varía el tamaño de la carga del barco. Se le da el valor de 204 kg que es el primer ejemplo.

- Se define la variable tiempo para las ecuaciones diferenciales y ver como evoluciona el sistema en el tiempo. Se le da un valor de $t = 0$ para que comience en el tiempo 0s de la gráfica.

- El diferencial de tiempo, ha de tomar un valor pequeño, así que se le da el valor $dt = 0,1s$ pero podría tomar cualquier otro también pequeño. Cuanto más pequeño sea el diferencial de tiempo más lento será el proceso de estabilización del sistema.

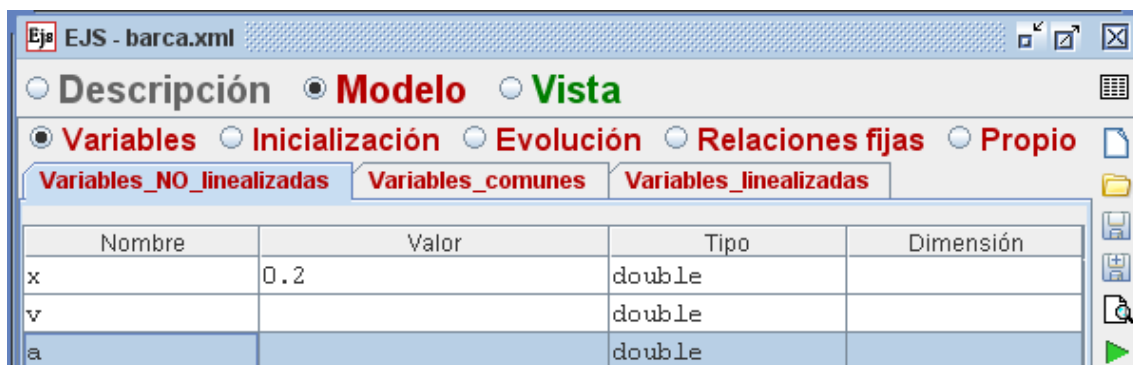
- La gravedad por definición es igual a $9,8 \text{ m/s}^2$.

- La constante k sirve para calcular la fuerza de rozamiento del agua sobre el barco, y en el enunciado especifica que $k = 1600 \text{ N*s/m}$.

- La densidad del agua es $\rho = 1000 \text{ kg/m}^3$.

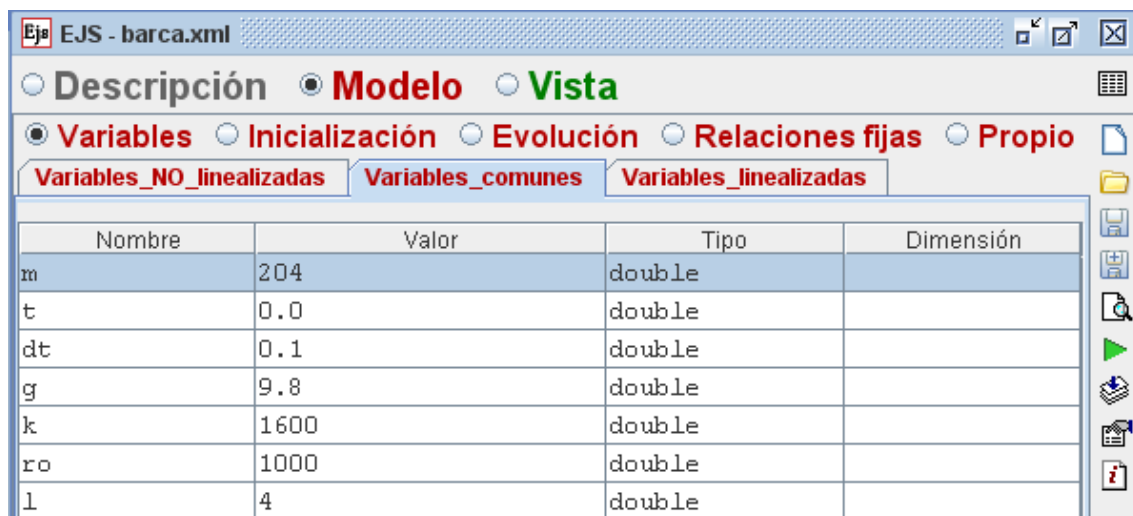
- La longitud del barco es $l = 4\text{m}$.

Hay que añadir que algunas de estas variables son exclusivas del modelo lineal y otras que son comunes con el modelo no lineal, por lo que se procede a agruparlas en 3 paginas de variables, las variables del modelo no lineal (ver imagen 34), las comunes (ver imagen 35) y las del modelo lineal.



| Nombre | Valor | Tipo | Dimensión |
|--------|-------|--------|-----------|
| x | 0.2 | double | |
| v | | double | |
| a | | double | |

Figura 34: variables no linealizadas



| Nombre | Valor | Tipo | Dimensión |
|--------|-------|--------|-----------|
| m | 204 | double | |
| t | 0.0 | double | |
| dt | 0.1 | double | |
| g | 9.8 | double | |
| k | 1600 | double | |
| ro | 1000 | double | |
| l | 4 | double | |

Figura 35: variables comunes

5.2.1.2 Ecuaciones

Una vez definidas las variables, dentro del panel modelo, se pasa a escribir las ecuaciones necesarias. Estas ecuaciones están definidas en el enunciado.

La ecuación del enunciado: $m(t)g - k\dot{x}(t) - (\rho g L / \sqrt{3})x(t)^2 = m(t)\ddot{x}(t)$, no se puede poner directamente en el programa ya que el programa no deja poner segundas derivadas, por lo que se procede a definir las dos ecuaciones diferenciales: velocidad y aceleración (ver figura 36), y una vez definidas velocidad y aceleración se pone la ecuación en relaciones fijas (ver figura 37).

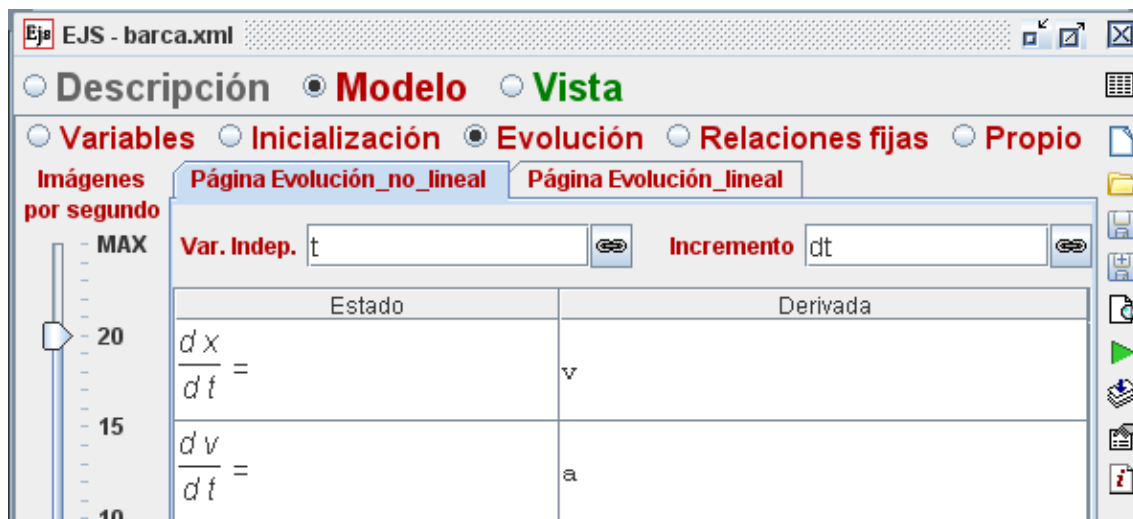


Figura 36: ecuaciones evolución

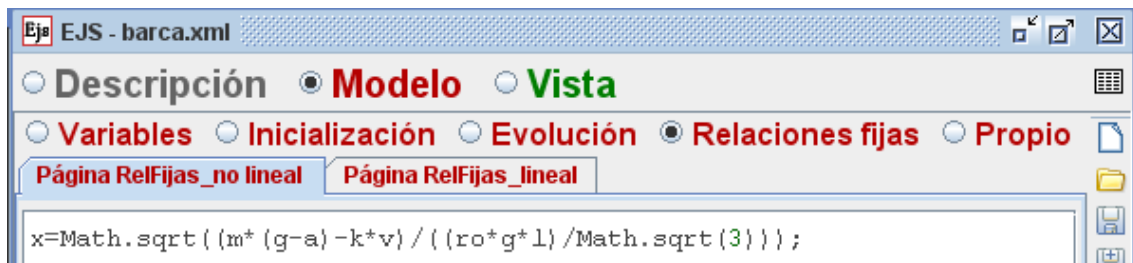


Figura 37: ecuación relación fija

Nota: a la hora de escribir las ecuaciones hay que dejar la variable a calcular despejada en el primer miembro de la ecuación para que el programa no de errores de compilación.

5.2.1.3 Vista

Una vez definidas las variables y las ecuaciones, hay que asociar esas variables a unas figuras que simulen esas variaciones de profundidad, esto se hace desde el panel Vista.

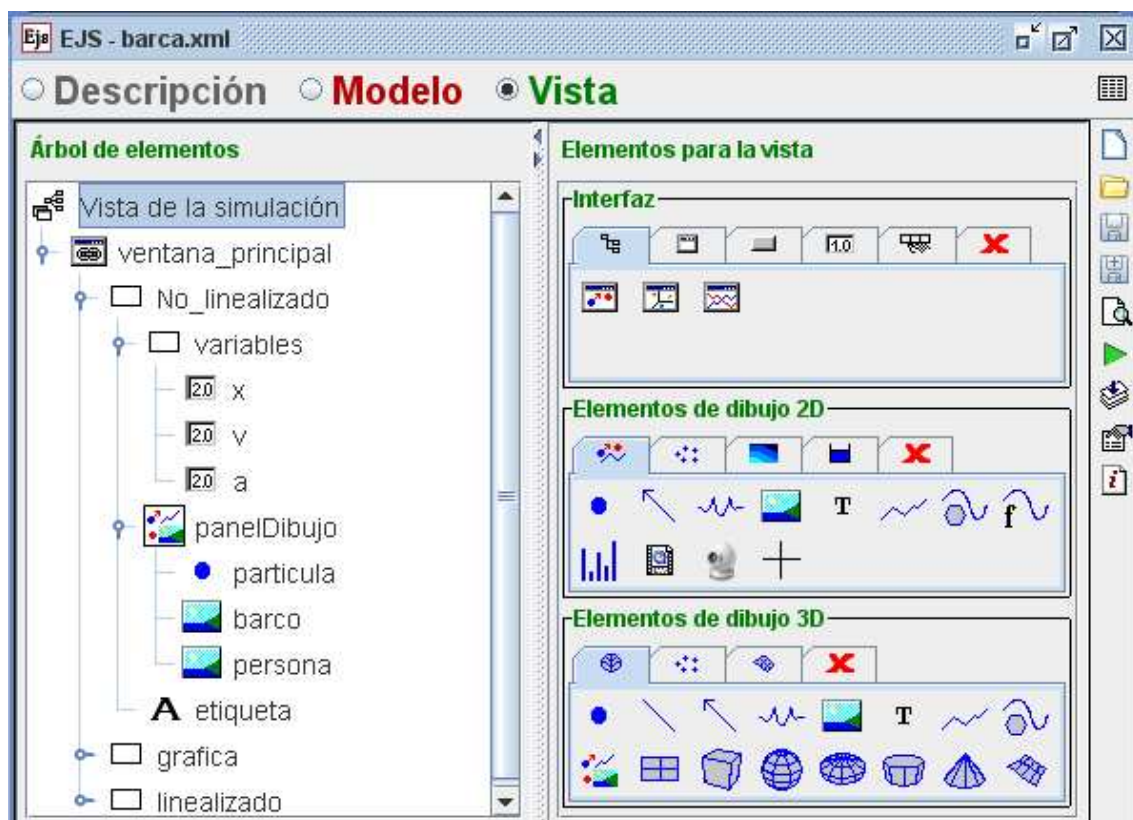


Figura 38: elementos de la vista

Lo primero es colocar la ventana principal, donde se colocará la gráfica de la evolución de la profundidad con el tiempo y el dibujo del depósito con las casillas de las

variables con los valores que van tomando. Cada dibujo o gráfica va colocado en un panel contenedor básico.

Dentro del contenedor que recibe el nombre de “No linealizado” se coloca:

- Un contenedor con las variables que pueden interesar como la profundidad, la velocidad o la aceleración.
- Un panel de dibujo donde se colocan los elementos necesarios para simular el barco, la carga y el mar.
- Una etiqueta para ponerle un título a la simulación, en este caso, diferenciar la simulación no linealizada de la linealizada.

Una vez colocados todos los elementos visuales se asocian las variables a los elementos:

Se puede observar en la figura 39 un ejemplo de cómo modificando las propiedades de la carga del barco (figura llamada persona) se asocia el desplazamiento, a lo largo del eje Y de la figura, con la variable profundidad (x); o el tamaño de la figura con la masa, la cual se divide entre un entero para que se pueda apreciar bien la figura y no sobresalga.

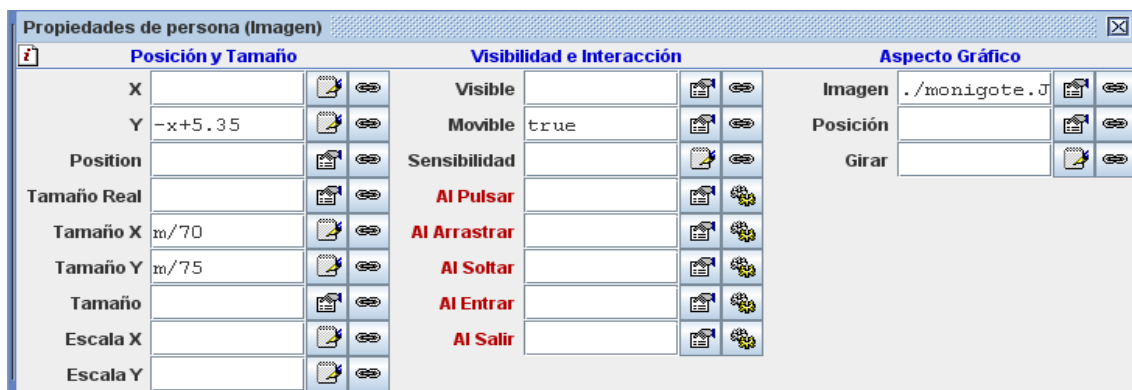


Figura 39: propiedades de la altura y tamaño de la carga

Una vez colocado y dimensionado el barco hay que incluir la gráfica, para ello dentro de la ventana principal se coloca un panel contenedor básico, al que se le llama gráfica y dentro de él se incluye:

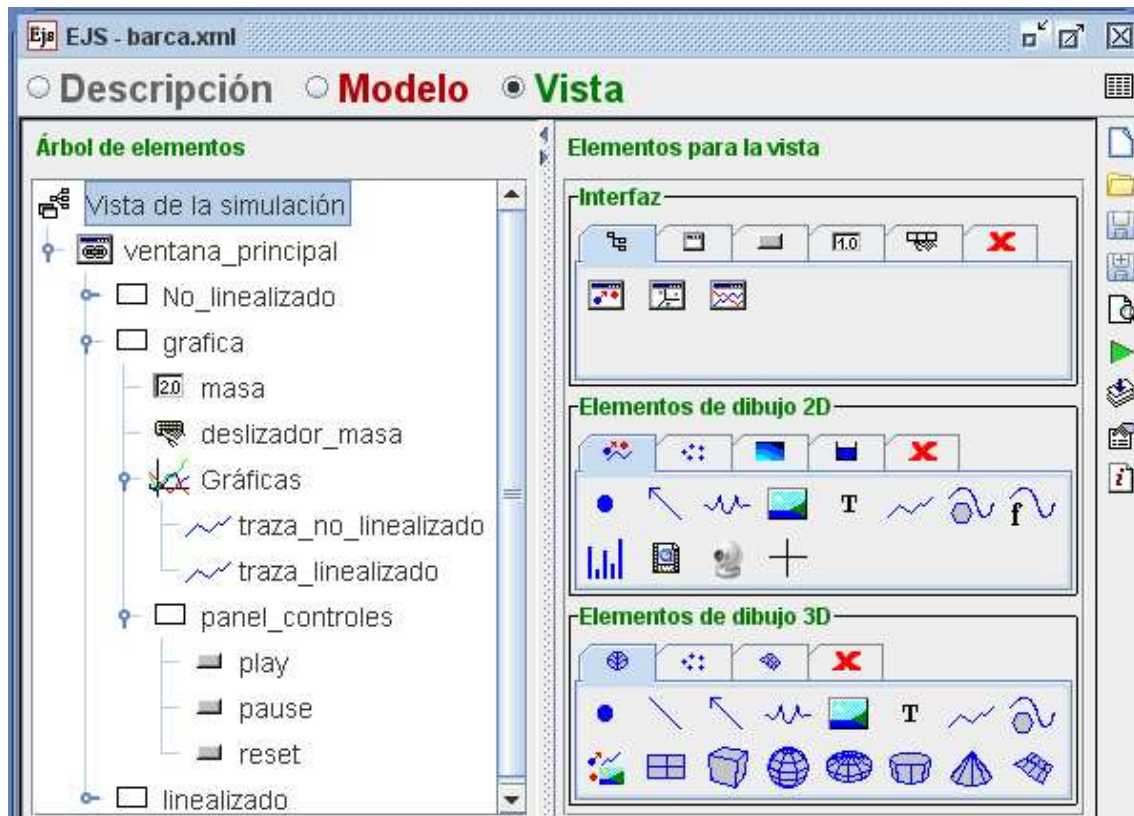


Figura 40: elementos de la gráfica

- Una casilla para informar numéricamente de la masa.
- Un deslizador para que el usuario pueda modificar la masa deslizando el ratón sobre la barra.
- Un panel de gráficas donde se incluyen tantas trazas como se necesiten (en este caso dos).
- Un panel de controles para controlar la simulación, con tres botones: play, pause y reset. Estos botones cumplen su función indicándoselo en el campo acción y el

campo activado para indicar cuando se encuentra activo dicho botón (ver figuras 41, 42 y 43).



Figura 41: propiedades del botón play

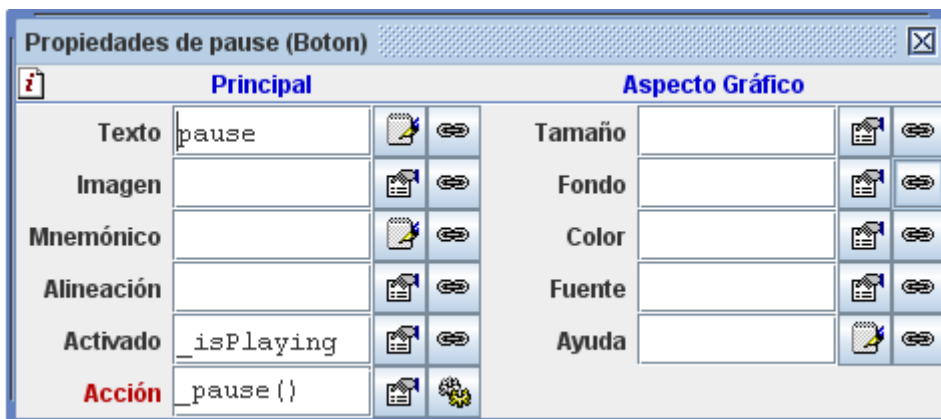


Figura 42: propiedades del botón pause

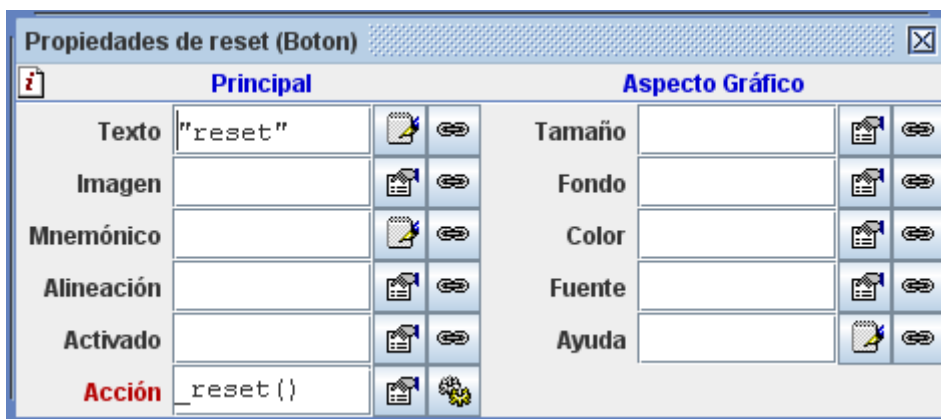


Figura 43: propiedades del botón reset

Ya se encontraría listo el modelo para ser simulado, pero lo que se pretende con este ejercicio es ver la comparación del sistema linealizado con el sistema no linealizado.

5.2.2 Modelo lineal

A partir del modelo no lineal se encuentra el punto donde la profundidad permanece constante, ese es el punto de equilibrio entorno al cual hay que linealizar.

5.2.2.1 Variables

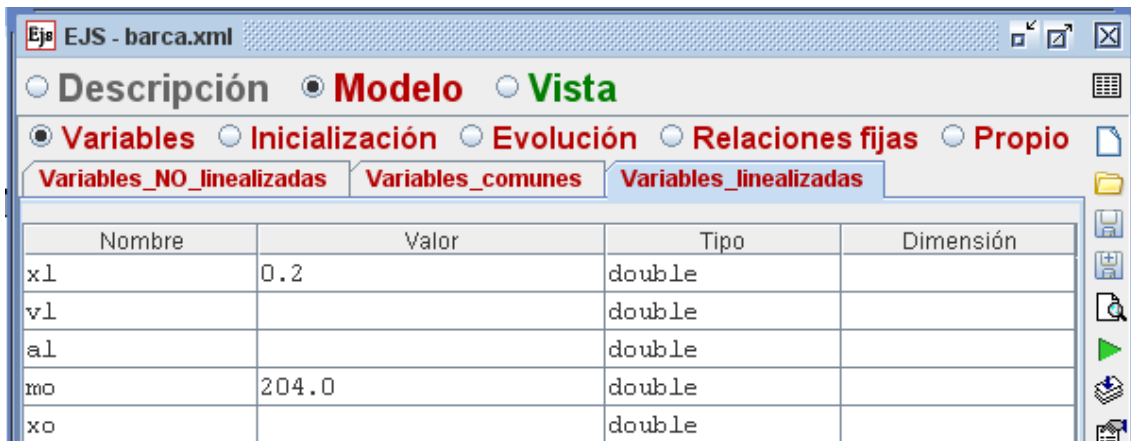
Hay que definir variables linealizadas y las variables que definen el punto de equilibrio.

Las variables linealizadas son:

- x_l : profundidad linealizada
- v_l : velocidad linealizada
- a_l : aceleración linealizada

Las variables que definen el punto de equilibrio son:

- m_0 : masa de equilibrio
- x_0 : profundidad de equilibrio



| Nombre | Valor | Tipo | Dimensión |
|--------|-------|--------|-----------|
| x1 | 0.2 | double | |
| v1 | | double | |
| a1 | | double | |
| mo | 204.0 | double | |
| xo | | double | |

Figura 44: variables linealizadas

El resto de variables (m, k, g, t, dt, ro, l) están definidas en el modelo no lineal, las cuales se pueden utilizar también para este modelo, son las llamadas variables comunes.

5.2.2.2 Ecuaciones

Ahora debemos tener en cuenta que la masa no es constante, se produce un escalón en la masa:



En equilibrio: $x = \text{cte}$, por tanto $\dot{x} = \ddot{x} = 0$, de donde:

$$m_0 g = \frac{\rho g L}{\sqrt{3}} x_0^2 \Rightarrow x_0 = 0.2972m$$

La ecuación del modelo lineal del sistema es:

$$mg - k \dot{x} - 2 \left(\frac{\rho g L}{\sqrt{3}} \right) x_0 x = m_0 \ddot{x}$$

Esta ecuación diferencial de segundo grado, se puede desarrollar como dos diferenciales de primer grado que definen la velocidad y la aceleración ya que el programa no permite poner segundas derivadas como se ha explicado anteriormente (ver figura 45), y una vez definidas velocidad y aceleración se define la ecuación en la pestaña de relaciones fijas (ver figura 46).

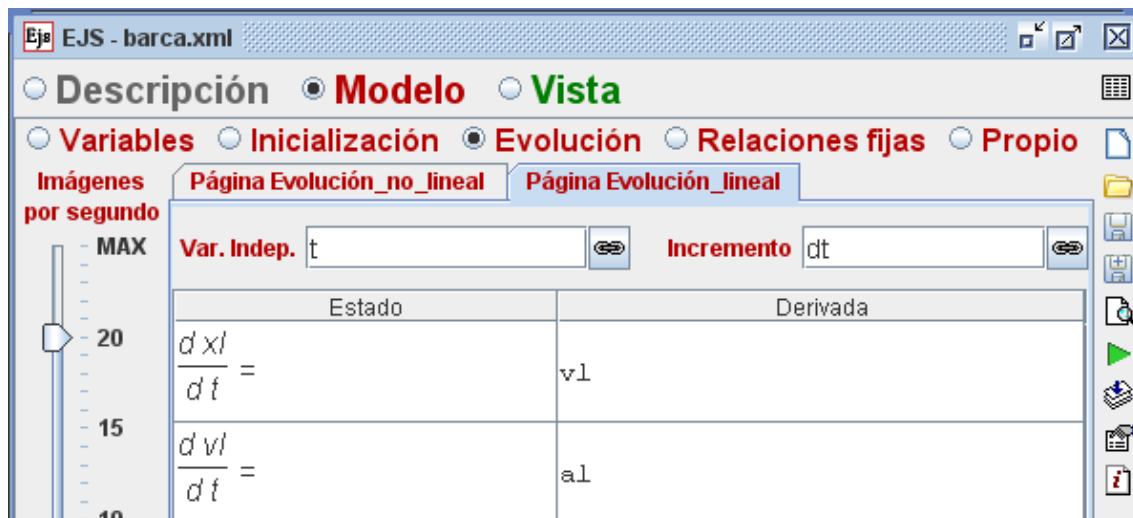


Figura 45: ecuación diferencial

También se define la profundidad de equilibrio:

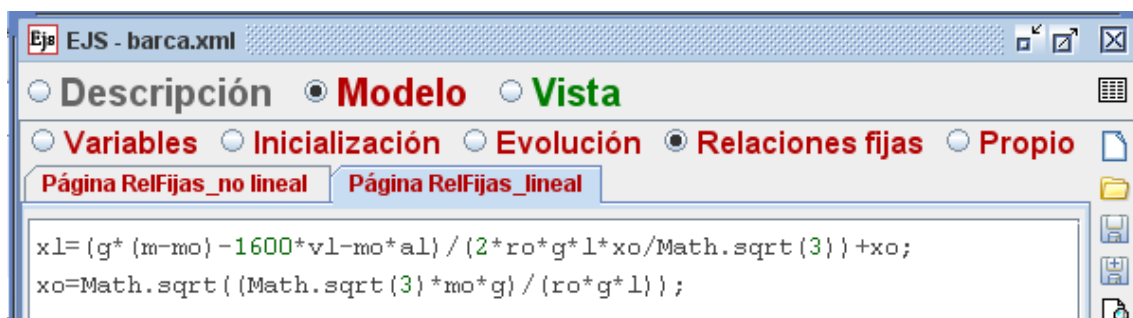


Figura 46: ecuaciones relaciones fijas

5.2.2.3 Vista

El desarrollo del modelo linealizado es el mismo proceso que el del no linealizado, la única diferencia consiste en el momento de enlazar las variables con los dibujos, que en este caso hay que unirlos con las variables linealizadas.



Figura 47: propiedades de la altura y tamaño de la carga

También se incluyen dos casillas donde se ve en torno a qué punto de equilibrio se linealiza (x_0 , m_0), pudiendo ser modificado este para ver los resultados que se obtienen linealizando en torno a distintos puntos de equilibrio.



Figura 48: elementos de la vista

Una vez realizado este trabajo solo queda ver la simulación:

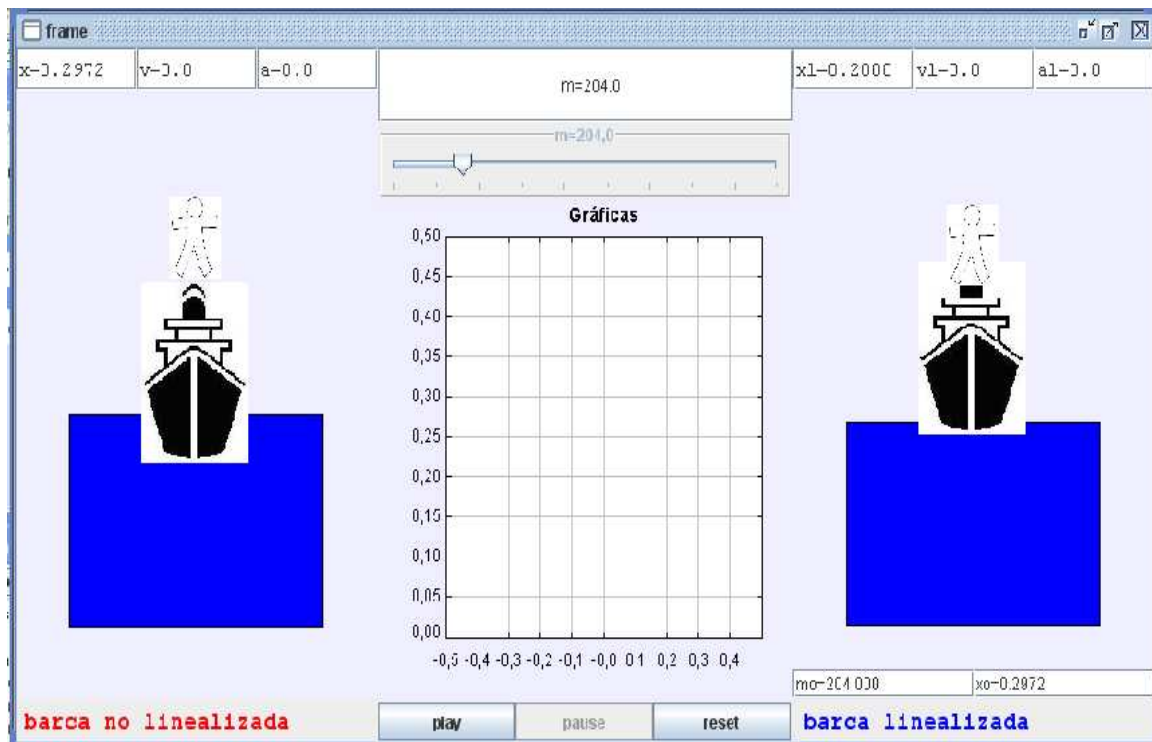


Figura 49: simulación final



Capítulo 6

Depósito controlado

6.1 Enunciado del problema

Dado el sistema de la figura representa un depósito de área constante A , alimentado por un caudal de entrada $q_e(t)$. El depósito pierde líquido con un caudal de salida $q_s(t)$.

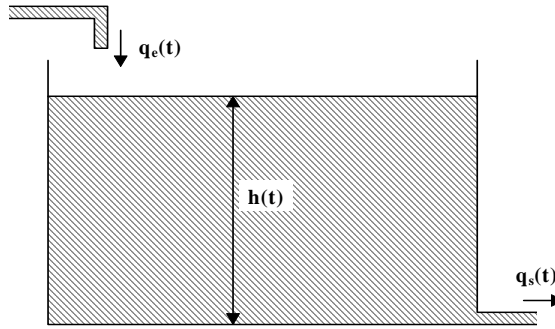


Figura 50: depósito

Se pretende mantener el nivel del depósito a la altura indicada por la referencia, utilizando un regulador proporcional, tal y como se muestra en la figura siguiente:

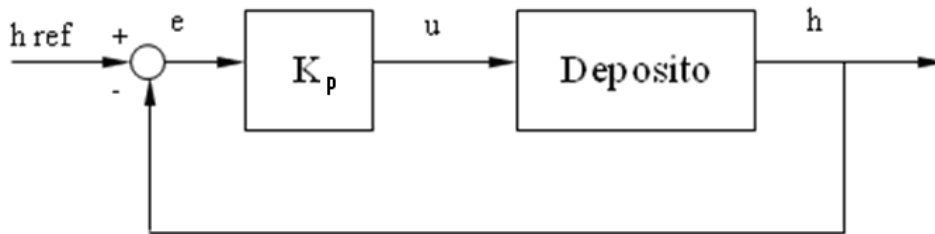


Figura 51: diagrama de bloques del regulador

Las ecuaciones del depósito ya han sido modeladas en el ejemplo de anterior. En este caso hay que añadir las ecuaciones necesarias para hacer el control:

$$q_e(t) - q_s(t) = A \frac{d h(t)}{dt} \quad (1)$$

$$q_s(t) = a \sqrt{2 \cdot g \cdot h(t)} \quad (2)$$

$$q_e(t) = k \cdot u(t) \quad (3)$$

Estas tres ecuaciones se resumen en la siguiente:

$$\frac{dh(t)}{dt} = -\frac{a}{A}\sqrt{2gh(t)} + \frac{k}{A}u(t) \quad (4)$$

Que es la ecuación que modela el sistema.

La ecuación que calcula el error entre la señal de salida y la referencia es:

$$error = h_{ref} - h$$

Y la acción de control que actúa sobre el depósito es:

$$u = Kp \cdot error$$

6.1 Resolución del problema

Conociendo el enunciado del problema se procede a dar una solución mediante la herramienta software EJS.

Hay que aclarar que se simularán dos modelos de barcos, un modelo no linealizado y otro linealizado. Se parte del modelo no linealizado, y con éste se obtiene el modelo linealizado para poder comparar los resultados.

6.1.1 Variables

Para comenzar se toma el modelo no lineal, lo primero es realizar la lista de variables dentro del panel modelo, las variables que se van a utilizar son:

- h : altura del líquido en el depósito
- h_{ref} : altura deseada del líquido en el depósito
- K_p : constante de proporcionalidad del regulador
- R : radio de la circunferencia que define el depósito
- A : área del depósito
- r : radio de la circunferencia que define las tuberías
- a : área de las tuberías
- g : gravedad
- k : constante
- t : tiempo
- dt : diferencial de tiempo
- error: el error cometido por el controlador
- u : acción de control

- A la altura se le da el valor por defecto de 0,16m para ver como se el regulador aumenta o disminuye el caudal de entrada para conseguir igualar la h_{ref} . Se le puede dar cualquier valor por defecto, el único interés de darle un valor distinto al valor de la h_{ref} es ver como se estabiliza el sistema.

- La variable h_{ref} toma el valor 0,1m pudiendo tomar cualquier valor entre 0 y 0,2m que se ha establecido. Se busca un valor diferente a los 0,16m de h para ver la respuesta del sistema en el arranque al tener los dos parámetros que compara con un valor diferente.

- La variable K_p toma un valor de 1000 por defecto ya que es el caso del primer ejemplo a realizar, pero puede tomar cualquier valor.

- El radio $R = 0,0295\text{m}$.

- El área del depósito es : $A = \pi \cdot R^2 = 2,73 \times 10^{-3} \text{ m}^2$.

- El radio $r = 0,0015\text{m}$.

- El área de las tuberías es : $a = \pi \cdot r^2 = 7,07 \times 10^{-6} \text{ m}^2$.

- La gravedad por definición es igual a $9,8 \text{ m/s}^2$.

- La constante k tiene un valor de $k = 2,7 \times 10^{-6}$.

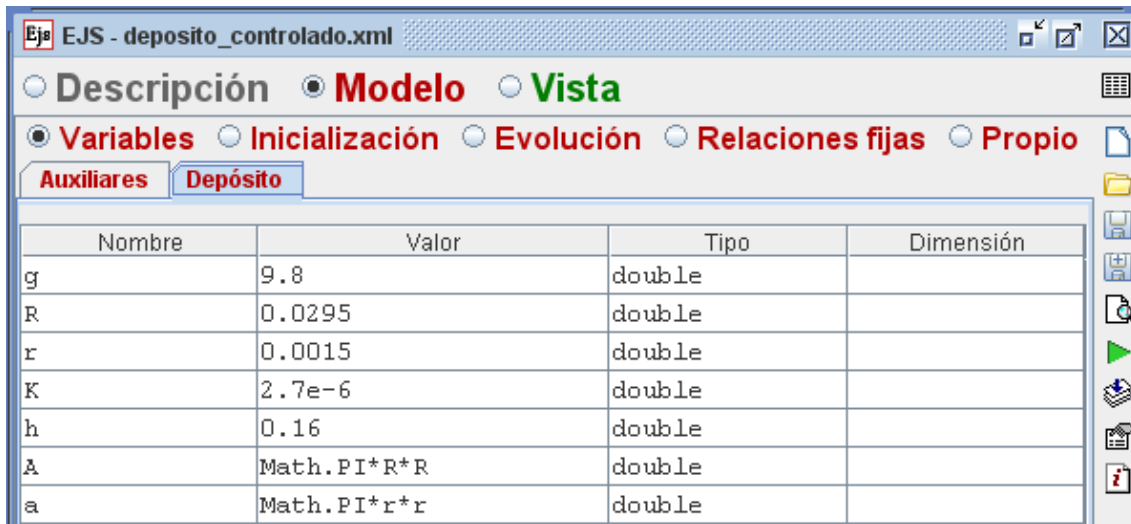
- Se define la variable tiempo para las ecuaciones diferenciales y ver como evoluciona el sistema en el tiempo. Se le da un valor de $t = 0$ para que comience en el tiempo 0s de la gráfica.

- El diferencial de tiempo, ha de tomar un valor pequeño, así que se le da el valor $dt = 0,1\text{s}$ pero podría tomar cualquier otro también pequeño. Cuanto más pequeño sea el diferencial de tiempo más lento será el proceso de estabilización del sistema.

- El error tiene un valor igual a 0, para que el programa cuando arranque calcule la diferencia de alturas y ofrezca un valor del error.

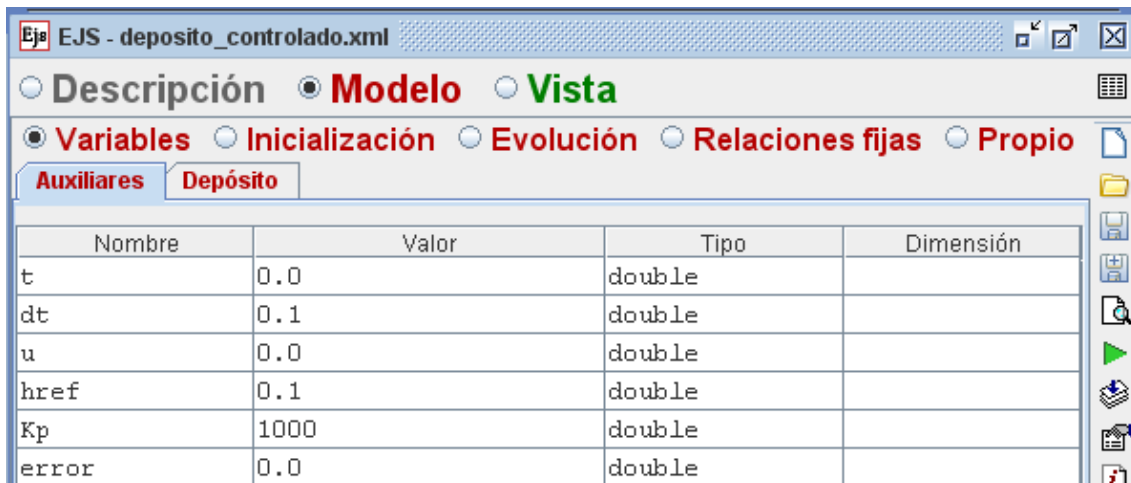
- La acción de control u tiene un valor inicial 0, para que cuando arranque, comience el programa a ejecutar la ecuación que le da valor al controlador y u tome el valor dado por el error.

Todas estas variables son agrupadas en dos pestañas, una la pestaña llamada depósito (ver figura 52) donde se encuentran las variables asociadas al depósito y otra pestaña la llamada auxiliares (ver figura 53) donde se encuentran las variables auxiliares para crear el modelo del controlador.



| Nombre | Valor | Tipo | Dimensión |
|--------|--------------------------|--------|-----------|
| g | 9.8 | double | |
| R | 0.0295 | double | |
| r | 0.0015 | double | |
| K | 2.7e-6 | double | |
| h | 0.16 | double | |
| A | $\text{Math.PI} * R * R$ | double | |
| a | $\text{Math.PI} * r * r$ | double | |

Figura 52: variables depósito



| Nombre | Valor | Tipo | Dimensión |
|--------|-------|--------|-----------|
| t | 0.0 | double | |
| dt | 0.1 | double | |
| u | 0.0 | double | |
| href | 0.1 | double | |
| Kp | 1000 | double | |
| error | 0.0 | double | |

Figura 53: variables auxiliares

6.1.2 Ecuaciones

Una vez definidas las variables, dentro del panel modelo, se pasa a escribir las ecuaciones necesarias. Estas ecuaciones están definidas en el enunciado.

La ecuación $\frac{dh(t)}{dt} = -\frac{a}{A}\sqrt{2gh(t)} + \frac{k}{A}u(t)$, se coloca en la pestaña de evolución

ya que es una ecuación diferencial (ver figura 54). Pero al ser el segundo miembro tan extenso y hacer uso de la propia h, se crea una función auxiliar llamada “ode” (ver figura 55) que calcula el segundo miembro de la ecuación, además del error y el parámetro u de control.

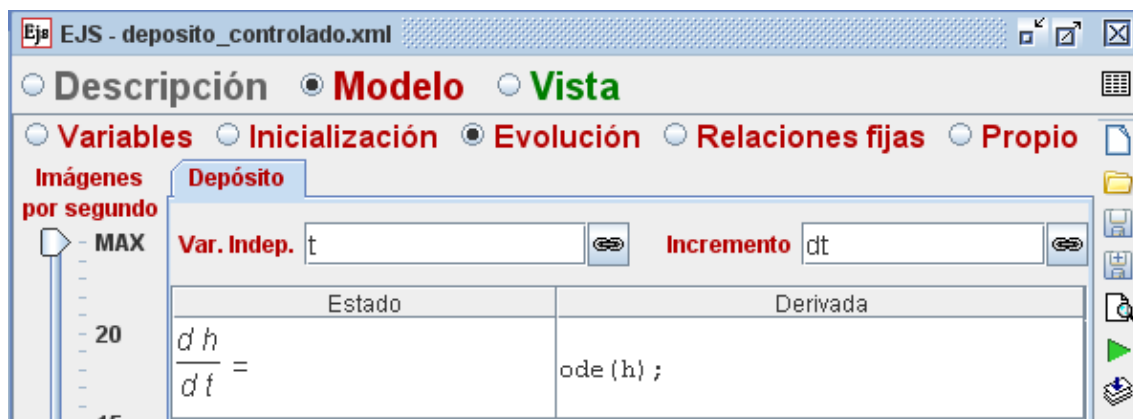


Figura 54: ecuación evolución

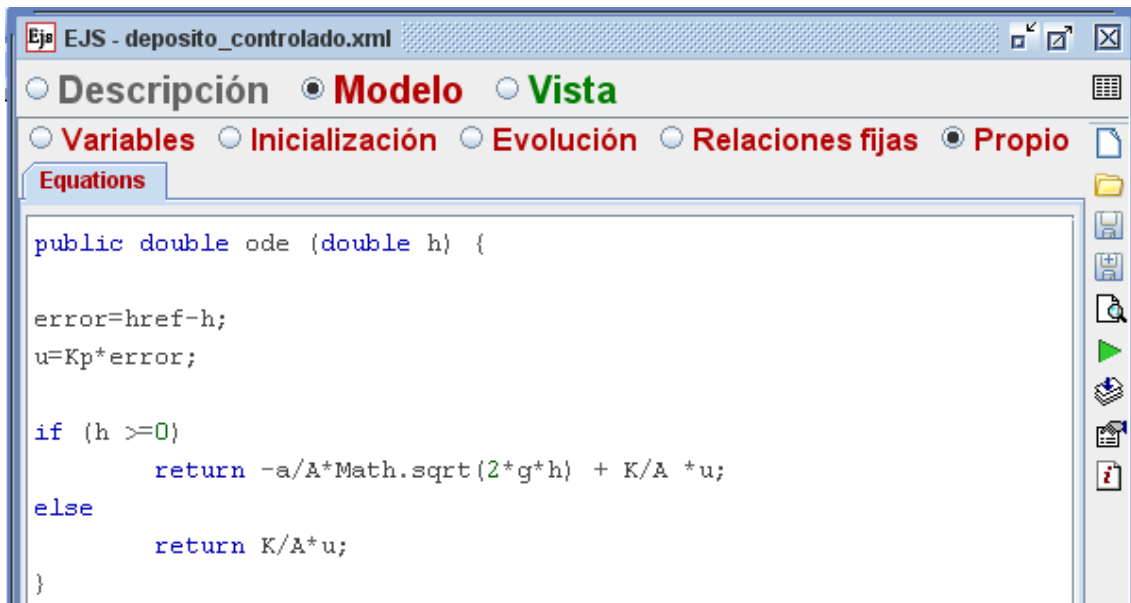


Figura 55: función código propio

También se añade otra sentencia en relaciones fijas (ver figura 56) en la que se establece que si la altura sobrepasa la altura del depósito (0.2) la acción de control sea 0.0, es decir que el controlador solo funcione cuando el caudal de agua se encuentre dentro de las dimensiones del depósito.

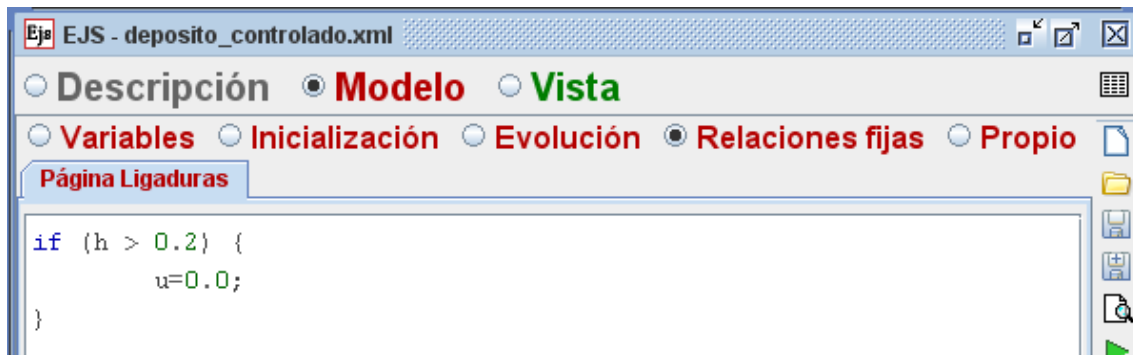


Figura 56: función relación fija

6.1.3 Vista

Una vez definidas las variables y las ecuaciones, hay que asociar esas variables a unas figuras que simulen esas variaciones de altura, esto se hace desde el panel Vista.

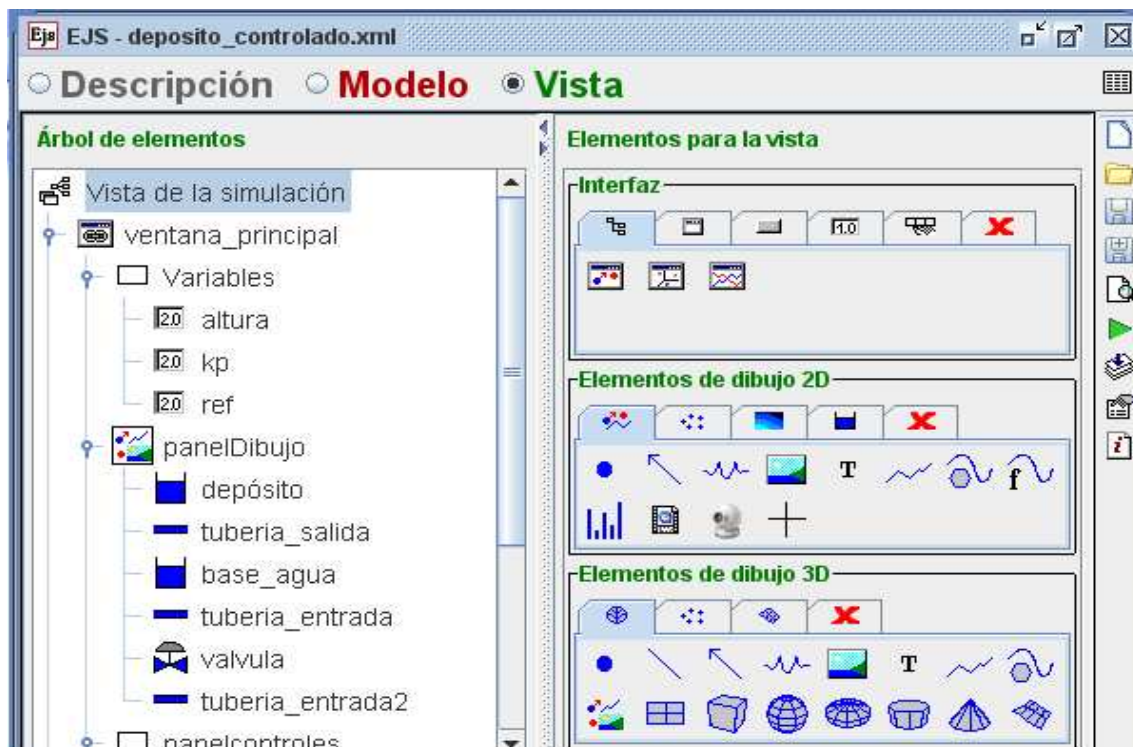


Figura 57: elementos de la vista I

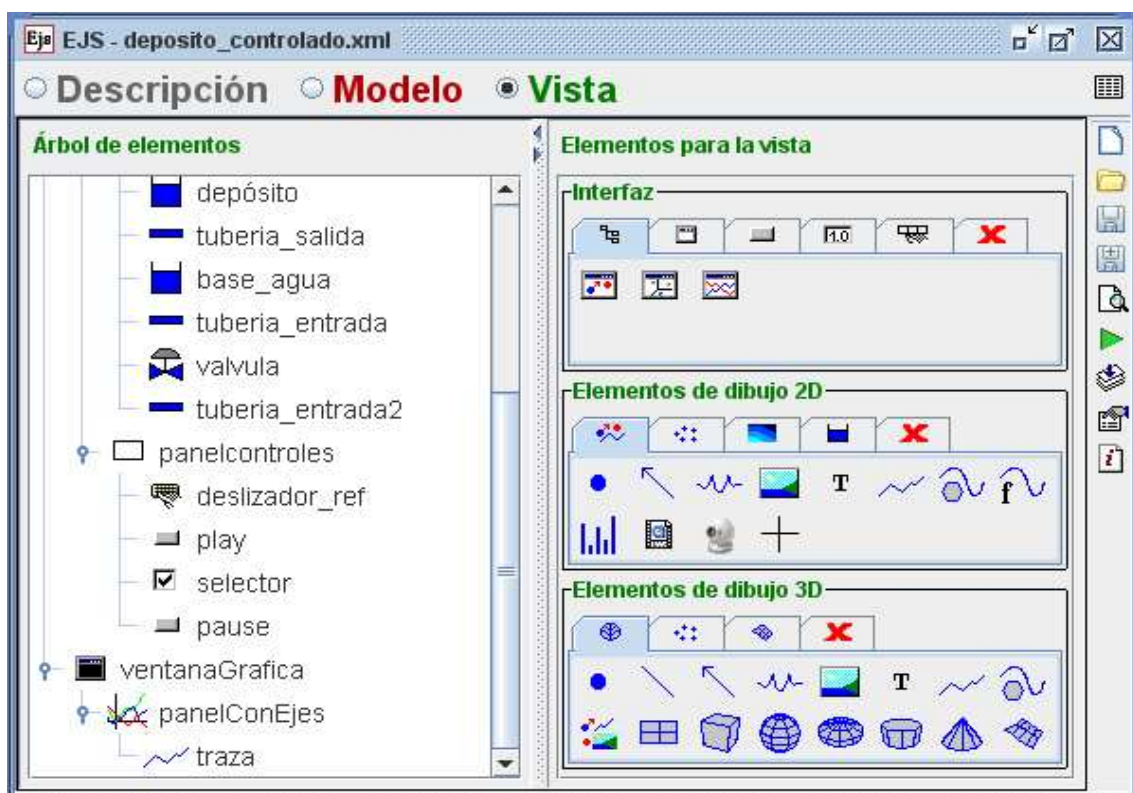


Figura 58: elementos de la vista II

Lo primero es colocar la ventana principal, donde se coloca el campo de las variables que muestran los valores que éstas van tomando, el panel de dibujo con el dibujo del depósito y el panel de controles que muestra los botones de control de la simulación. En este caso la gráfica que muestra la evolución de la altura real (h) con el tiempo se coloca separada de la ventana principal por su menor interés.

Dentro de estos campos se colocan las casillas o elementos de la vista necesarios para simular el depósito controlado:

- El contenedor con las variables muestra las tres variables que más interesantes pueden resultar como la altura real (h), la altura buscada (h_{ref}) y la variable proporcional que regula el control (K_p).
- El panel de dibujo donde se colocan los elementos necesarios para simular el contenedor y las tuberías.
- El panel de controles dispone de un deslizador para que el usuario pueda cambiar la altura h_{ref} y ver como el regulador automáticamente modifica el caudal de entrada a la magnitud necesaria para alcanzar una altura aproximada a la buscada (la aproximación dependerá del valor dado a K_p), un botón de puesta en marcha de la simulación (play), otro para pausar la simulación (pause) y una casilla que ofrece la posibilidad de cerrar o volver a colocar la ventana de la gráfica.

Como se ha dicho anteriormente, separado de la ventana principal se coloca el panel de la ventana gráfica, donde se coloca un panel con ejes y se incluye la traza de la altura real.

Una vez colocados todos los elementos visuales se asocian las variables a los elementos:

Se puede observar en la figura 59 un ejemplo de cómo modificando las propiedades del depósito se asocia el nivel con la altura (h), lo cual va a simular como sube y baja en nivel del agua.

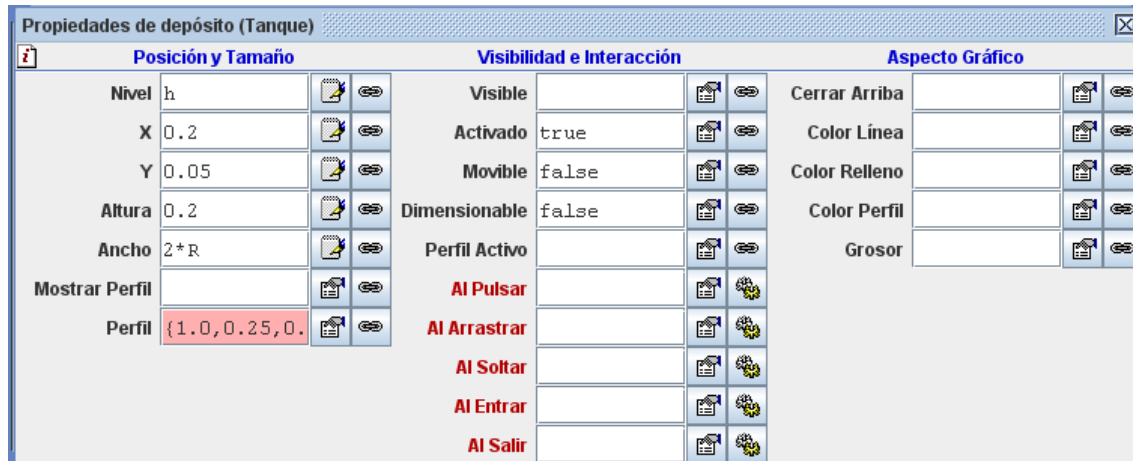


Figura 59: propiedades del depósito

Dentro de la ventana de propiedades se modifican los atributos del dibujo para acercarlo más al ejercicio en particular.

Los botones del panel de controles cumplen su función indicándoselo en el campo acción y el campo activado para indicar cuando se encuentra activo dicho botón (ver figuras 60 y 61).



Figura 60: propiedades del botón play



Figura 61: propiedades del botón pause

El botón selector está asociado a la gráfica mediante una variable creada (mostrargrafi) que relaciona el botón selector con la gráfica (ver figura 62).

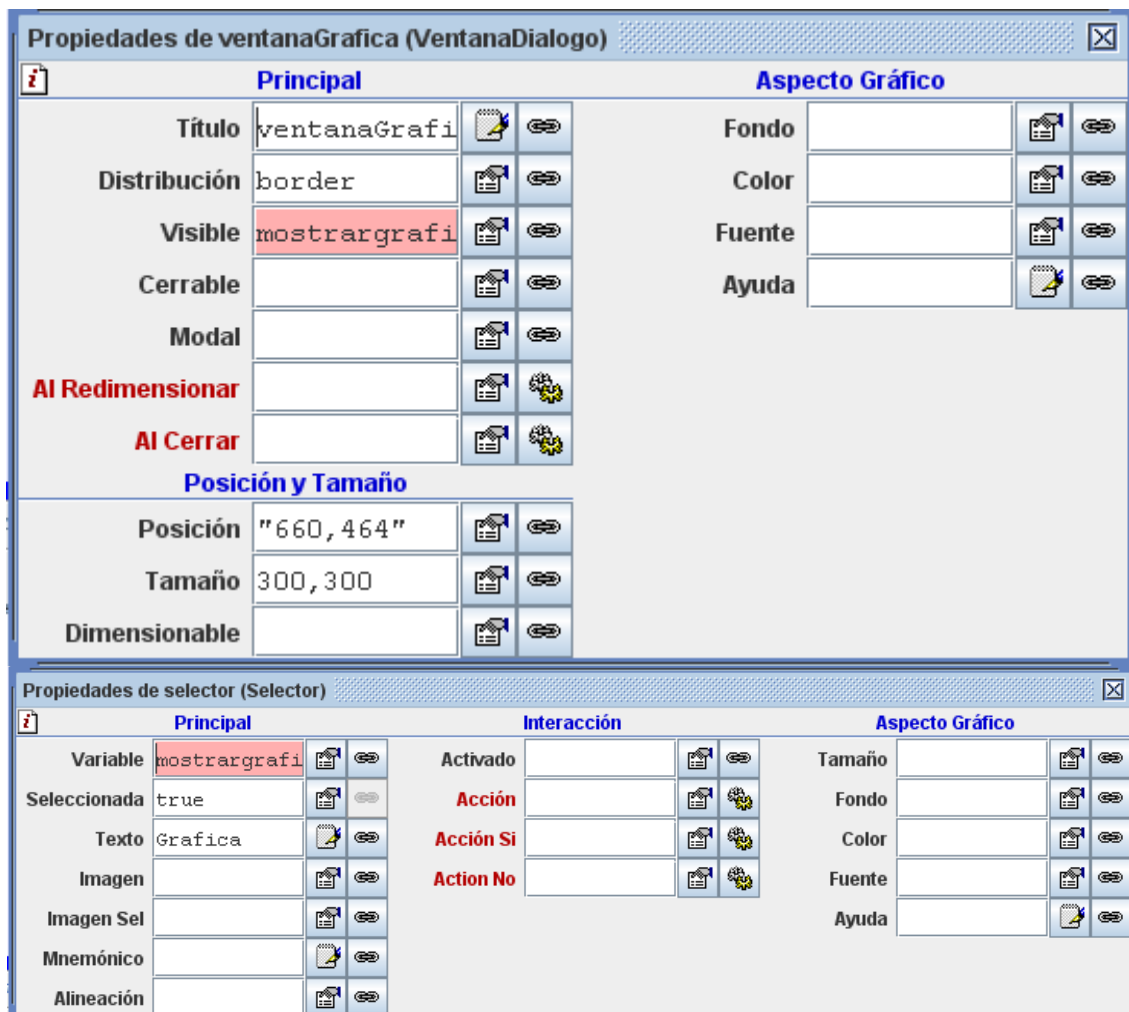


Figura 62: propiedades de la ventana grafica y el botón selector

Ya se encontraría listo el modelo para ser simulado, por tanto sólo queda ver la simulación:

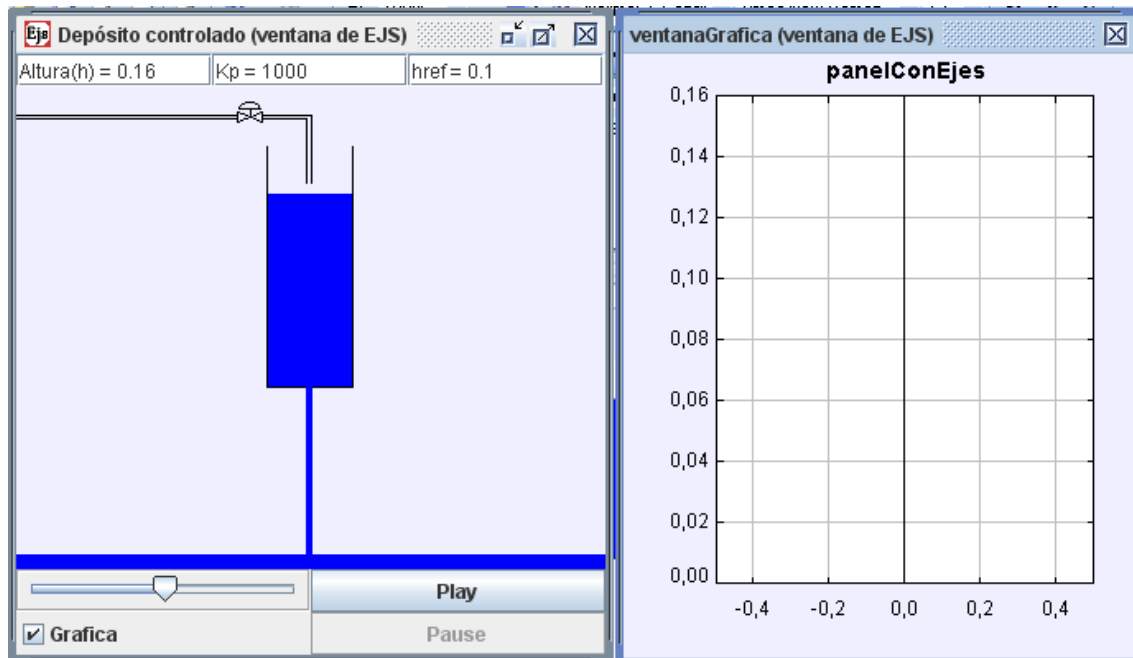


Figura 63: simulación final



Capítulo 7

Casos prácticos

7.1 Depósito

7.1.1 Incremento en el caudal de la entrada de 1 a 1,1 m³/s

En el primer punto de funcionamiento que se estudia, el caudal de entrada toma el valor $q_{11} = 1 \text{ m}^3/\text{s}$. Para este caudal de entrada, el sistema debe alcanzar un punto de equilibrio y se estabilizará en una altura h_1 , la cual se determina a partir de la expresión (3) sustituyendo los datos y constantes conocidos:

$$h_1 = \left(\frac{1}{2}\right)^2 = 0,25m$$

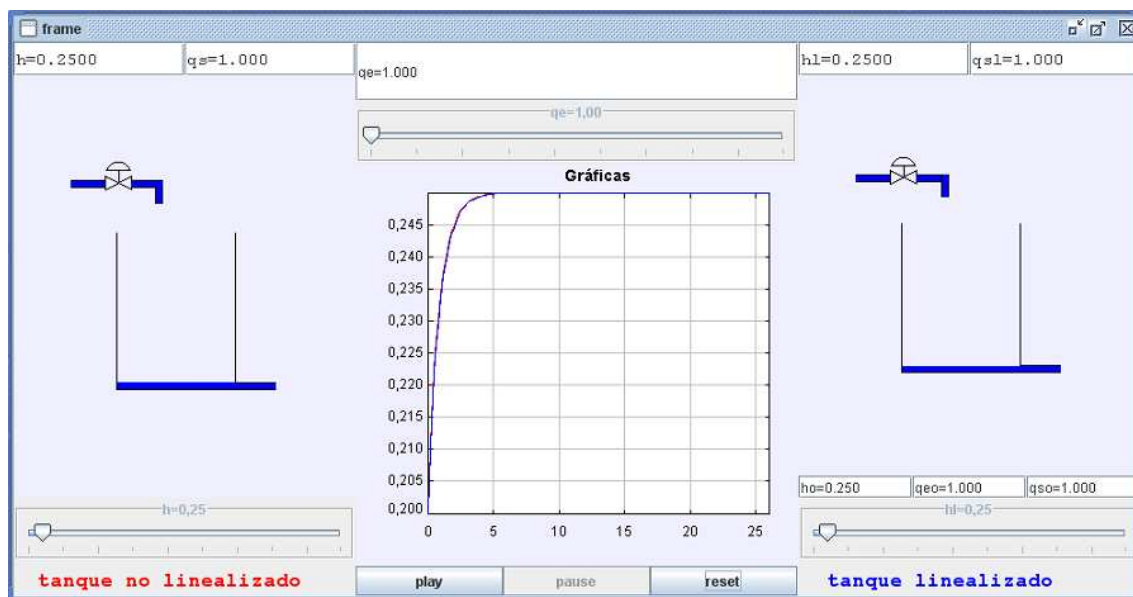


Figura 64: caudal de entrada 1m³/s

Si partiendo del equilibrio, se produce una pequeña variación en el caudal de entrada, pasando éste a ser $q_{12} = 1,1 \text{ m}^3/\text{s}$, el sistema evolucionará hasta un nuevo punto de equilibrio, puesto que el sistema es estable. Para el caso el sistema real la nueva altura de equilibrio vendrá dada por la ecuación (3):

$$h_2 = \left(\frac{1,1}{2} \right)^2 = 0,3025m$$

Y para el sistema linealizado, la variación de altura se calcula con (6).

$$\Delta h_{12} = \frac{2 \cdot \sqrt{0,25}}{2} \cdot (1,1 - 1) = 0,05m$$

Con lo que la altura de equilibrio, de acuerdo con (7), será:

$$h_2 = h_1 + \Delta h_{12} = 0,25 + 0,05 = 0,3m$$

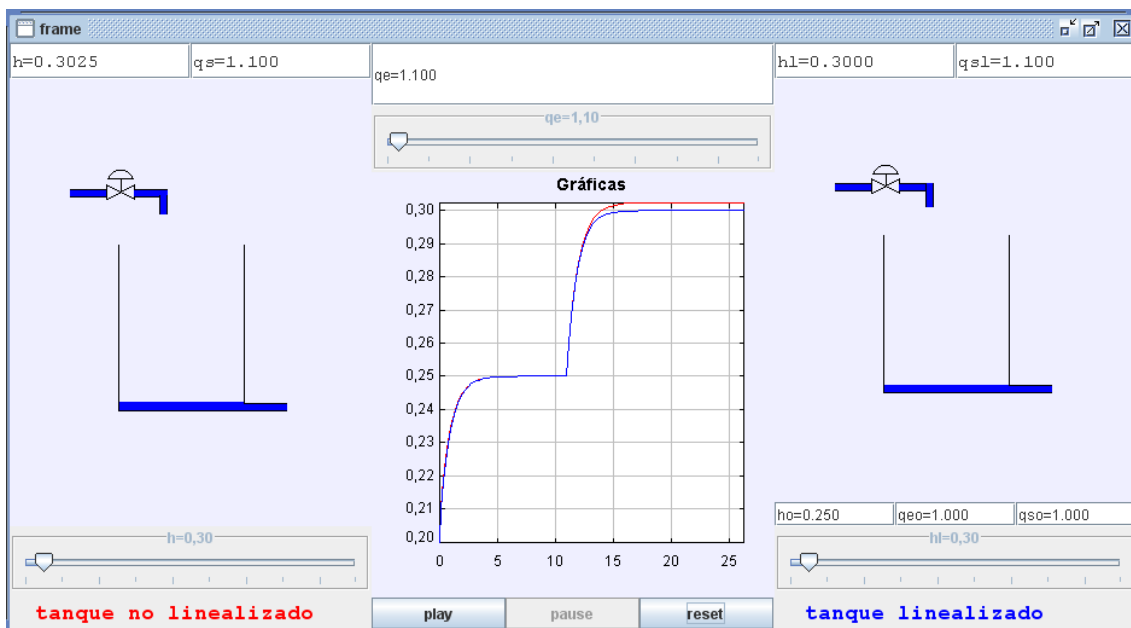


Figura 65: caudal de entrada $1,1 \text{ m}^3/\text{s}$

-El incremento de altura producido en el sistema no lineal es de 25cm a 30cm.

-En el sistema lineal la altura del depósito pasa de 25cm a 30,25cm.

-El error producido en la linealización es de 0,25cm (30,25cm-30cm), que comparado con los 30cm de altura real, se puede considerar los 0,25cm como un error pequeño:

$$\frac{0,25}{30} \cdot 100 = 0,83\% , \text{ por lo que la linealización es válida.}$$

➤ Esto ocurre así porque se estudia la respuesta de un sistema en un punto cercano al punto de equilibrio, por lo que el sistema lineal ofrece una respuesta muy parecida al sistema real.

7.1.2 Incremento en el caudal de la entrada de 1 a 4,5 m³/s

El punto de funcionamiento es el mismo que en el caso precedente ya que se parte del mismo caudal de entrada $q_{21} = 1 \text{ m}^3/\text{s}$. Por tanto la altura h_1 del líquido en el depósito en este punto según la ecuación (3) es:

$$h_1 = \left(\frac{1}{2}\right)^2 = 0,25m$$

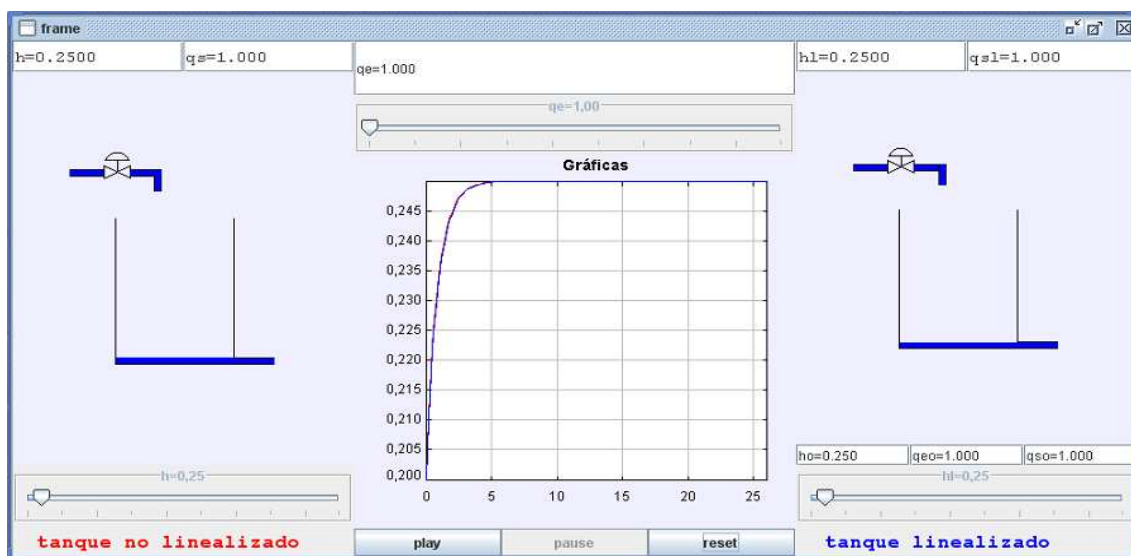


Figura 66: caudal de entrada 1,1 m³/s

En este caso la variación producida en el caudal de entrada es mucho mayor, siendo el caudal final $q_{22} = 4,5 \text{ m}^3/\text{s}$. Cuando el sistema alcance un nuevo punto de equilibrio, la altura a la que se estabilizará el sistema real viene dada por la expresión (3):

$$h_3 = \left(\frac{4,5}{2} \right)^2 = 5,0625 \text{ m}$$

Mientras que para el sistema linealizado, se produce una variación de altura que de acuerdo con (6) valdrá:

$$\Delta h_{13} = \frac{2 \cdot \sqrt{0,25}}{2} \cdot (4,5 - 1) = 1,75 \text{ m}$$

Con lo que la altura de equilibrio, de acuerdo con (7), será:

$$h_3 = h_1 + \Delta h_{13} = 0,25 + 1,75 = 2 \text{ m}$$

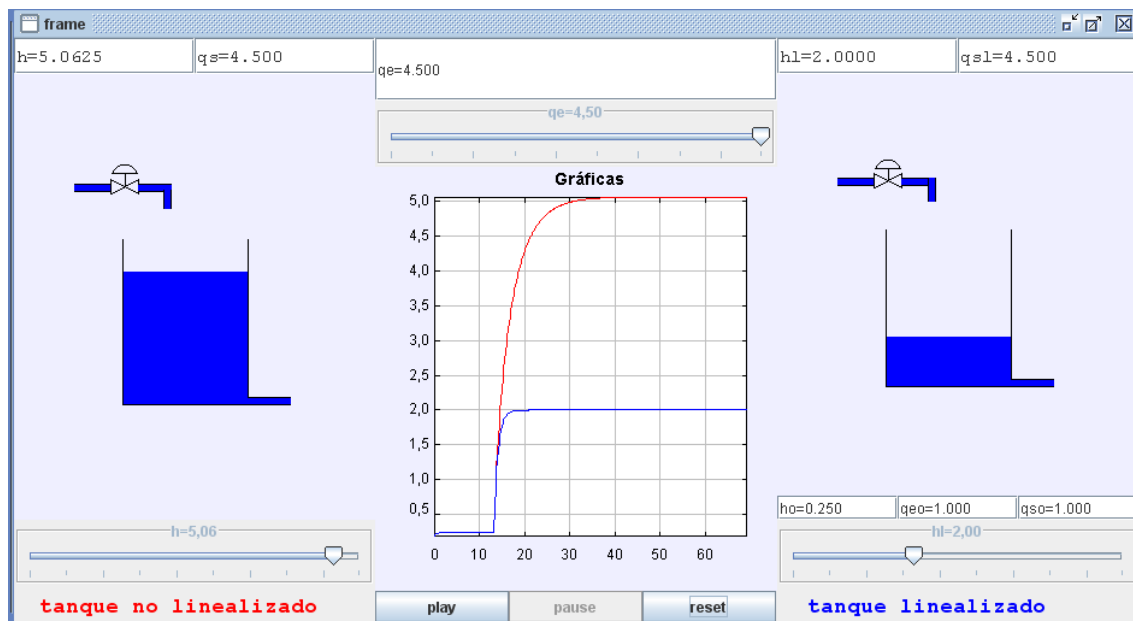


Figura 67: caudal de entrada $4,5 \text{ m}^3/\text{s}$

-El incremento de altura producido en el sistema no lineal es de 25cm a 5,0625m.

-En el sistema lineal la altura del depósito pasa de 25cm a 2m.

-El error producido en la linealización es de 3,0625m (5,0625m-2m), que comparado con los 5,0625m de altura real, se puede considerar los 3,0625m como un error muy grande: $\frac{3,0625}{5,0625} \cdot 100 = 60,49\%$, por lo que la linealización no es válida.

➤ En este caso la diferencia es significativa, lo que indica que el sistema linealizado no se comporta de manera adecuada ya que la variación de caudal es grande y nos alejamos mucho del punto de funcionamiento. La linealización no es válida, por lo que habrá que buscar un punto de funcionamiento próximo al del equilibrio final. El sistema lineal sólo sirve para pequeños incrementos de la variable de entrada.

7.1.3 Incremento en el caudal de la entrada de 4 a 4,5 m³/s

A diferencia de los casos anteriores en los cuales se partía de un mismo punto de funcionamiento en el que el caudal de entrada era $q_{11} = q_{21} = 1 \text{ m}^3/\text{s}$, en este tercer caso se parte de un caudal $q_{31} = 4 \text{ m}^3/\text{s}$. Esto quiere decir que se linealizará en un punto de funcionamiento distinto, con lo cual se obtendrá un sistema linealizado diferente al de los casos anteriores. Para este caudal de entrada, el sistema se estabilizará en una altura h_4 , calculada a partir de la expresión (3) sustituyendo los datos y constantes conocidos:

$$h_1 = \left(\frac{4}{2}\right)^2 = 4m$$

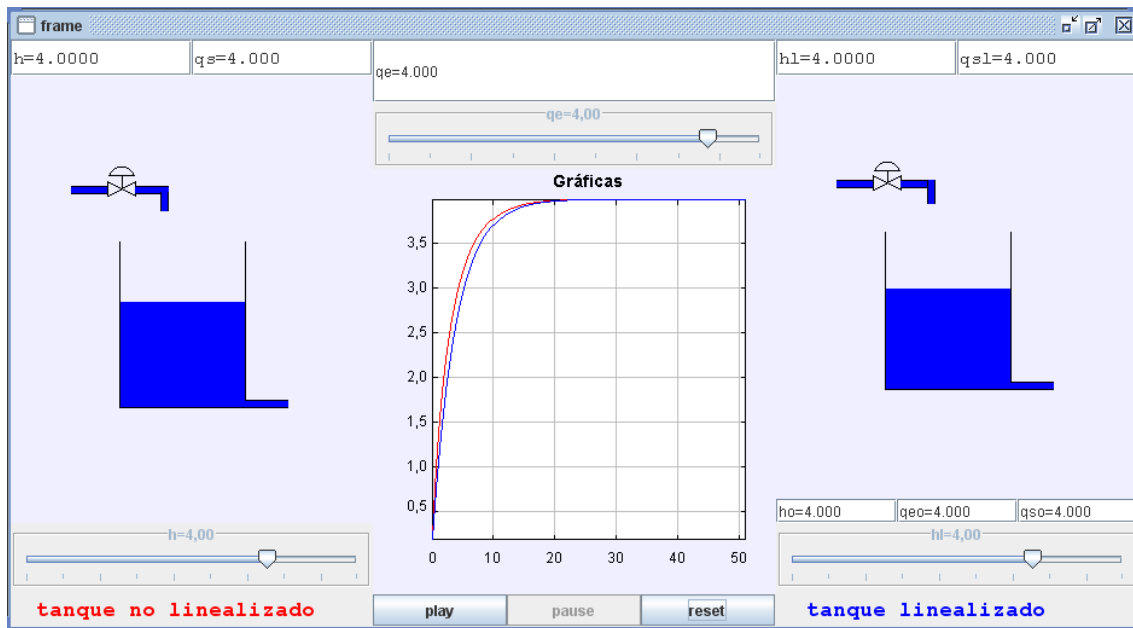


Figura 68: caudal de entrada $4 \text{ m}^3/\text{s}$

Si partiendo del equilibrio, se produce una pequeña variación en el caudal de entrada, pasando éste a ser $q_{22} = 4,5 \text{ m}^3/\text{s}$, el sistema evolucionará hasta un nuevo punto de equilibrio, puesto que el sistema es estable. Para el caso del sistema real la nueva altura de equilibrio vendrá dada por la ecuación (3):

$$h_5 = \left(\frac{4,5}{2} \right)^2 = 5,0625 \text{ m}$$

Vemos que la altura calculada coincide con la altura del apartado anterior. Esto es así porque en ambos casos se trata del sistema real para un mismo caudal de entrada de $4,5 \text{ m}^3/\text{s}$. El sistema lineal varía dependiendo del punto de funcionamiento escogido, pero el sistema es el mismo en todos los casos. Para el sistema linealizado, la variación de altura se calcula con (6).

$$\Delta h_{15} = \frac{2 \cdot \sqrt{4}}{2} \cdot (4,5 - 4) = 1 \text{ m}$$

Con lo que la altura de equilibrio, de acuerdo con (7), será:

$$h_5 = h_1 + \Delta h_{15} = 4 + 1 = 5m$$

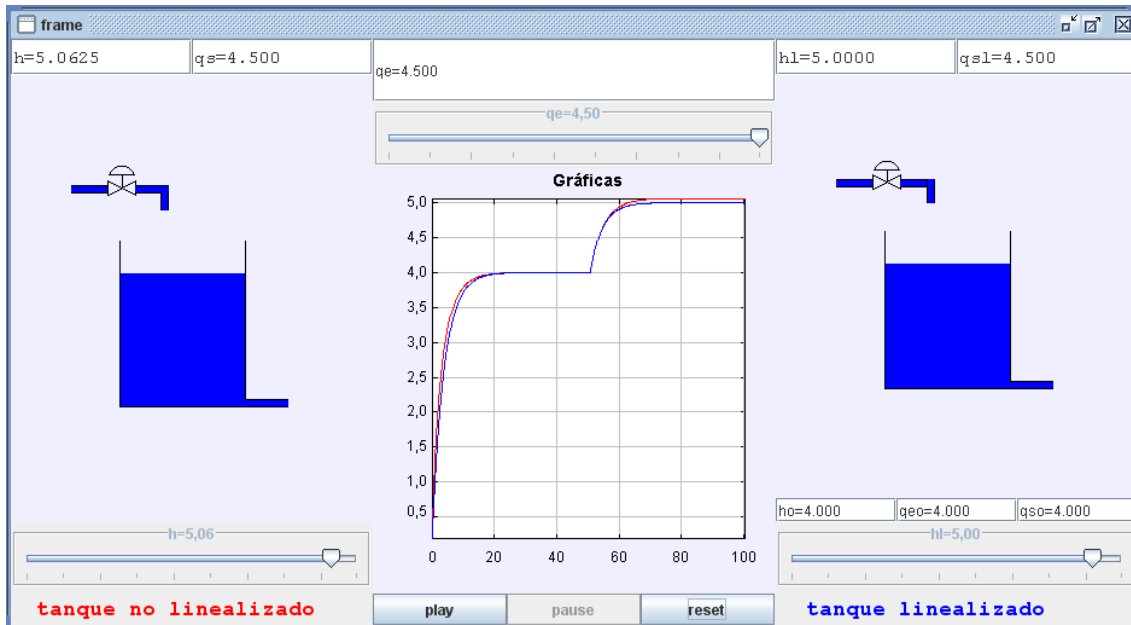


Figura 69: caudal de entrada 4,5 m³/s

-El incremento de altura producido en el sistema no lineal es de 4m a 5,0625m.

-En el sistema lineal la altura del depósito pasa de 4m a 5m.

-El error producido en la linealización es de 0,0625m (5,0625m-5m), que comparado con los 5,0625m de altura real, se puede considerar los 0,0625m como un error pequeño: $\frac{0,0625}{5,0625} \cdot 100 = 1,23\%$, por lo que la linealización es válida.

➤ En este caso al igual que el primer ejemplo, el incremento de caudal es pequeño, por lo que el sistema se aleja poco del punto de funcionamiento; por tanto al linealizar cerca del punto de equilibrio la respuesta lineal es parecida a la real.

7.2 Barco

7.2.1 Masa 204 kg

Ahora hay que tener en cuenta que la masa no es constante, en el primer punto de funcionamiento que se estudia, la masa toma el valor $m_{11} = 204$ kg. Para esta masa, el sistema debe alcanzar un punto de equilibrio y se estabilizará en una altura x_1 , la cual se determina a partir de la expresión $m(t)g - k\dot{x}(t) - (\rho g L / \sqrt{3})x(t)^2 = m(t)\ddot{x}(t)$ sustituyendo los datos y constantes conocidos.

En equilibrio: $x = \text{cte}$, por tanto $\dot{x} = \ddot{x} = 0$, de donde:

$$m_0 g = \frac{\rho g L}{\sqrt{3}} x_0^2 \Rightarrow x_0 = 0.2972m$$

La ecuación del modelo lineal del sistema es:

$$mg - k\dot{x} - 2\left(\frac{\rho g L}{\sqrt{3}}\right)x_0 x = m_0 \ddot{x}$$

La cual despejada para incluirla en el programa queda:

$$xl = \frac{(m - m_0) \cdot g - k \cdot vl - m_0 \cdot al}{\frac{2 \cdot ro \cdot g \cdot l \cdot x_0}{\sqrt{3}}} + x_0$$

Al linealizar en torno al punto de equilibrio en el que la masa es de 204 kg, el sistema linealizado tiene la misma respuesta. Ya que en el numerador de la ecuación

tenemos una diferencia de masas $m - m_0 = 0$, y una velocidad y aceleración igual a 0, por tanto:

$$x_l = x_0 = 0.2972m$$

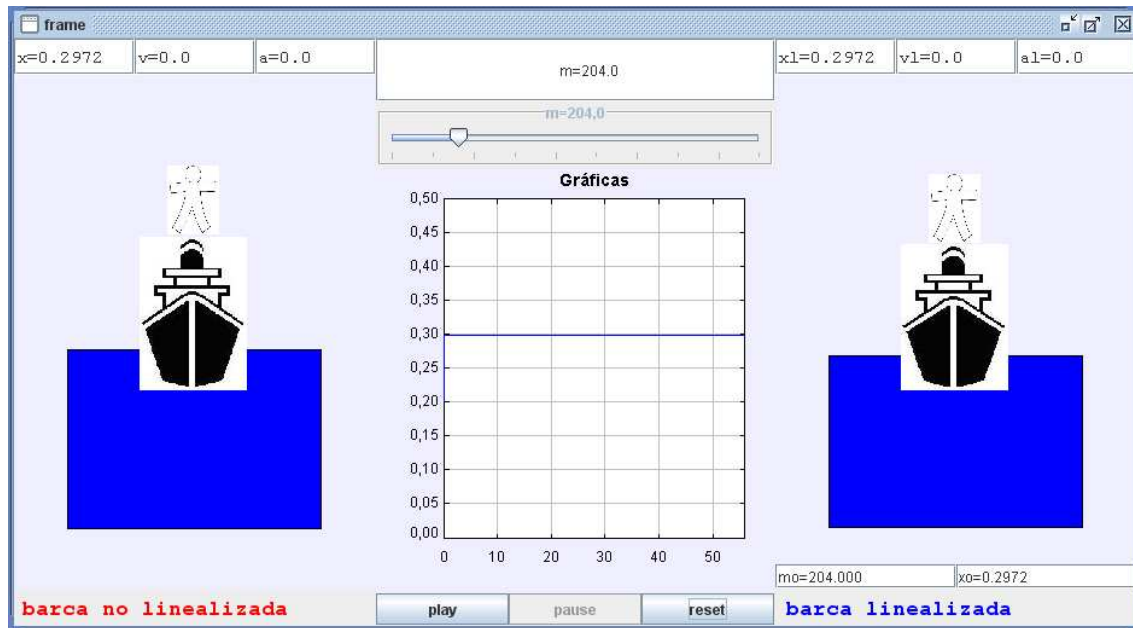


Figura 70: masa de 204 kg

El error producido es: $error = x_l - x_0 = 0.2972 - 0.2972 = 0m$, esto es así porque se estudia el sistema en el punto de equilibrio, por tanto el sistema lineal se comporta igual que el real.

7.2.2 Masa 304 kg

Después se produce un escalón en la masa de 100 kg, pero el punto de equilibrio se mantiene con la masa de 204 kg:

$$\begin{array}{l} \text{204 Kg} \end{array} \quad \begin{array}{l} \text{204+100 = 304 Kg} \end{array}$$

Para esta masa, el sistema debe estabilizar en una altura x_1 , la cual se determina a partir de la expresión $m(t)g - k\dot{x}(t) - (\rho g L / \sqrt{3})x(t)^2 = m(t)\ddot{x}(t)$ sustituyendo los datos y constantes conocidos.

En equilibrio: $x = \text{cte}$, por tanto $\dot{x} = \ddot{x} = 0$, de donde:

$$m_1 \cdot g = \frac{\rho g L}{\sqrt{3}} x_1^2 \Rightarrow x_1 = 0.3628m$$

La ecuación del modelo lineal del sistema es:

$$mg - k\dot{x} - 2\left(\frac{\rho g L}{\sqrt{3}}\right)x_0 x = m_0 \ddot{x}$$

La cual despejada para incluirla en el programa queda:

$$xl = \frac{(m - m_0) \cdot g - k \cdot vl - m_0 \cdot al}{\frac{2 \cdot ro \cdot g \cdot l \cdot x_0}{\sqrt{3}}} + x_0$$

Al linealizar en torno a un punto distinto del punto de equilibrio en el que la masa es de 204 kg, el sistema linealizado no tiene la misma respuesta. Sustituyendo cada variable por su valor se obtiene:

$$xl = 0.3701m$$

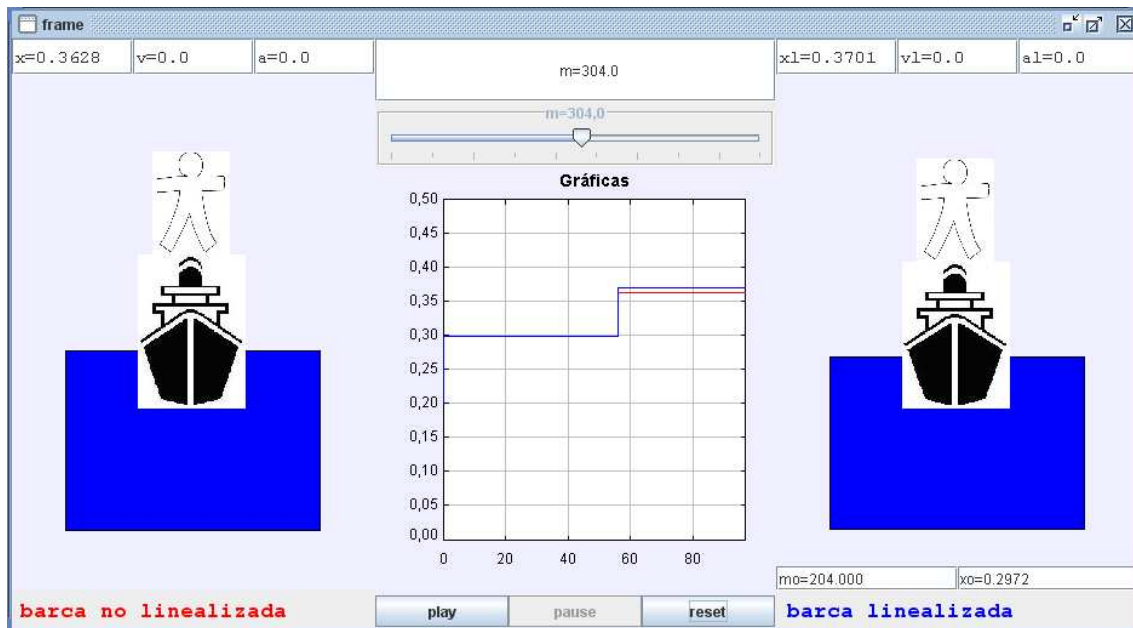
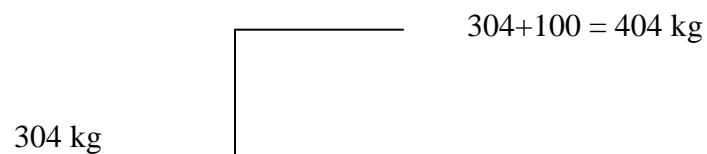


Figura 71: masa de 304 kg

El error producido es: $error = x1 - x0 = 0.3701 - 0.3628 = 0.0073m = 7.3mm$, esto es así porque se estudia el sistema en un punto distinto al punto de equilibrio, por tanto el sistema lineal no se comporta igual que el real, tiene un pequeño error.

7.2.3 Masa 404 kg

Más tarde se produce un escalón en la masa de 100 kg, pero el punto de equilibrio se mantiene con la masa de 204 kg:



Para esta masa, el sistema debe estabilizará en una altura x_2 , la cual se determina a partir de la expresión $m(t)g - k\dot{x}(t) - (\rho g L / \sqrt{3})x(t)^2 = m(t)\ddot{x}(t)$ sustituyendo los datos y constantes conocidos.

En equilibrio: $x = \text{cte}$, por tanto $\dot{x} = \ddot{x} = 0$, de donde:

$$m_2 \cdot g = \frac{\rho g L}{\sqrt{3}} x_2^2 \Rightarrow x_2 = 0.4183m$$

La ecuación del modelo lineal del sistema es:

$$mg - k\dot{x} - 2\left(\frac{\rho g L}{\sqrt{3}}\right)x_0 x = m_0 \ddot{x}$$

La cual despejada para incluirla en el programa queda:

$$xl = \frac{(m - m_0) \cdot g - k \cdot vl - m_0 \cdot al}{\frac{2 \cdot ro \cdot g \cdot l \cdot x_0}{\sqrt{3}}} + x_0$$

Al linealizar en torno a un punto distinto del punto de equilibrio en el que la masa es de 204 kg, el sistema linealizado no tiene la misma respuesta. Sustituyendo cada variable por su valor se obtiene:

$$xl = 0.4429m$$

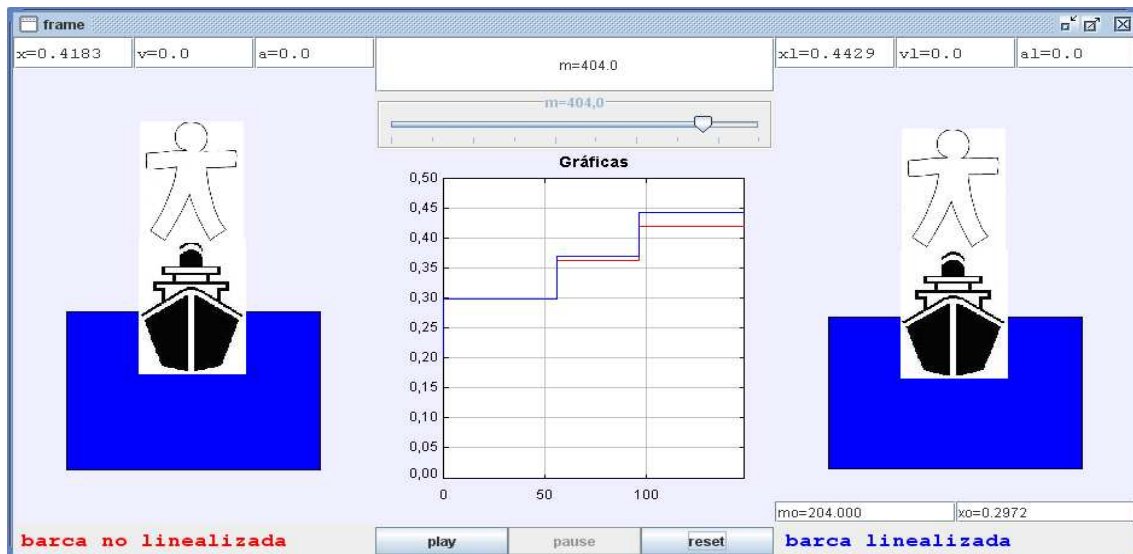


Figura 72: masa de 404 kg

El error producido es: $error = x_1 - x_0 = 0.4429 - 0.4183 = 0.0246m = 24.6mm$, esto es así porque se estudia el sistema en un punto más alejado del punto de equilibrio incluso que en el caso anterior, por tanto el sistema lineal no se comporta igual que el real, tiene un pequeño error.

➤ Como se puede observar para el ejemplo de masa de 204 kg el sistema lineal ofrece una respuesta igual al sistema real. Pero en cuanto la masa aumenta, y se va alejando de esa masa de equilibrio la respuesta se observa como va siendo peor. Cuanto más se aleja del punto de equilibrio el error cometido es mayor.

7.3 Depósito controlado

7.3.1 Ejercicio 1: $h_{ref} = 0.149m$ y $K_p = 1000$

Enunciado: tomar como altura de referencia $h_{ref} = 0.149$. Tomar $K_p = 1000$. Ver si el sistema es capaz de seguir a la entrada. ¿Cuál es el error?

Mediante la ecuación: $\frac{dh(t)}{dt} = -\frac{a}{A}\sqrt{2gh(t)} + \frac{k}{A}u(t)$ se despeja h . Esto se hace sustituyendo $u(t)$ por: $u = K_p \cdot error = K_p \cdot (h_{ref} - h)$ y conociendo que en equilibrio: $\frac{dh(t)}{dt} = 0$. Con todo esto la ecuación queda:

$$0 = -\frac{a}{A}\sqrt{2gh} + \frac{k}{A} \cdot K_p \cdot (h_{ref} - h)$$

Así despejando, resulta $h = 0.145m$, la cual coincide con el valor obtenido en la simulación.

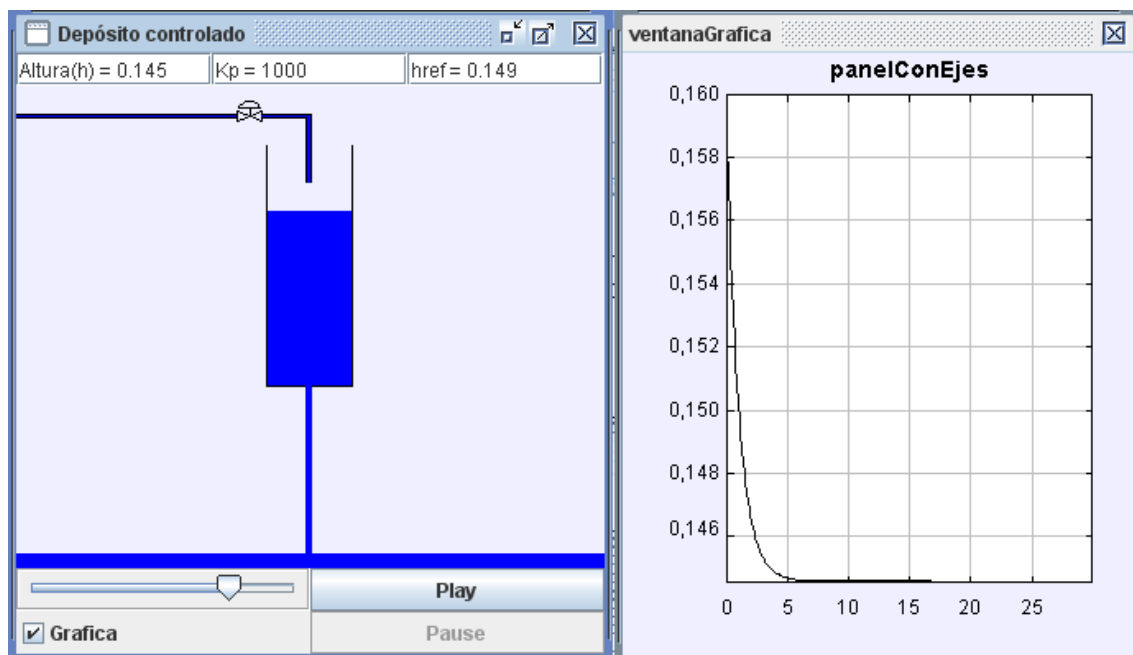


Figura 73: href = 0,149m y Kp = 1000

Se observa como el sistema es capaz de seguir a la entrada (href) pero no lo hace a la perfección, es decir comete un pequeño error, ese error es la diferencia de alturas, es decir, $error = 0.149 - 0.145 = 0.004m$, como se aprecia un error muy pequeño, por tanto el sistema para una $K_p = 1000$ sigue la entrada de forma correcta pero no exacta.

7.3.2 Ejercicio 2: $h_{ref} = 0.149m$ y $K_p = 500$

Enunciado: tomar como altura de referencia $h_{ref} = 0.149$. Tomar $K_p = 500$. Ver si el sistema es capaz de seguir a la entrada. ¿Cuál es el error?

Como antes se ha dicho, en equilibrio la ecuación queda así:

$$0 = -\frac{a}{A}\sqrt{2gh} + \frac{k}{A} \cdot K_p \cdot (h_{ref} - h)$$

Así despejando, resulta $h = 0.140m$, la cual coincide con el valor obtenido en la simulación.

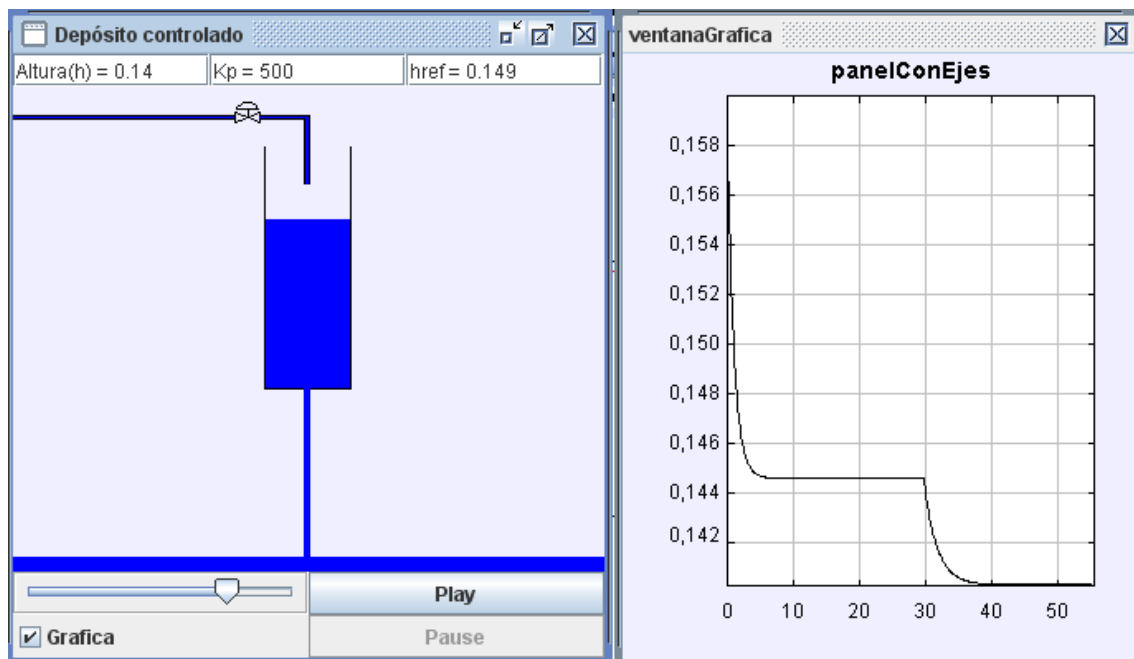


Figura 74: $h_{ref} = 0.149m$ y $K_p = 500$

Se observa como el sistema es capaz de seguir a la entrada (h_{ref}) pero no lo hace a la perfección, de hecho lo hace peor que cuando $K_p = 1000$, es decir comete un error mayor cuando el valor de K_p es menor, ese error es la diferencia de alturas, es decir, $error = 0.149 - 0.140 = 0.009m$, como se aprecia un error pequeño, por tanto el sistema

para una $K_p = 500$ sigue la entrada de forma correcta pero no exacta y su comportamiento es peor que el del ejemplo anterior, por tanto, se observa que cuanto menor es K_p , mayor es el error.

7.3.3 Ejercicio 3: $h_{ref} = 0.149\text{m}$ y $K_p = 1$

Enunciado: tomar como altura de referencia $h_{ref} = 0.149$. Tomar $K_p = 1$. Ver si el sistema es capaz de seguir a la entrada. ¿Qué está ocurriendo?

Como antes se ha dicho, en equilibrio la ecuación queda así:

$$0 = -\frac{a}{A}\sqrt{2gh} + \frac{k}{A} \cdot K_p \cdot (h_{ref} - h)$$

Así despejando, resulta $h = 0\text{m}$, la cual coincide con el valor obtenido en la simulación.

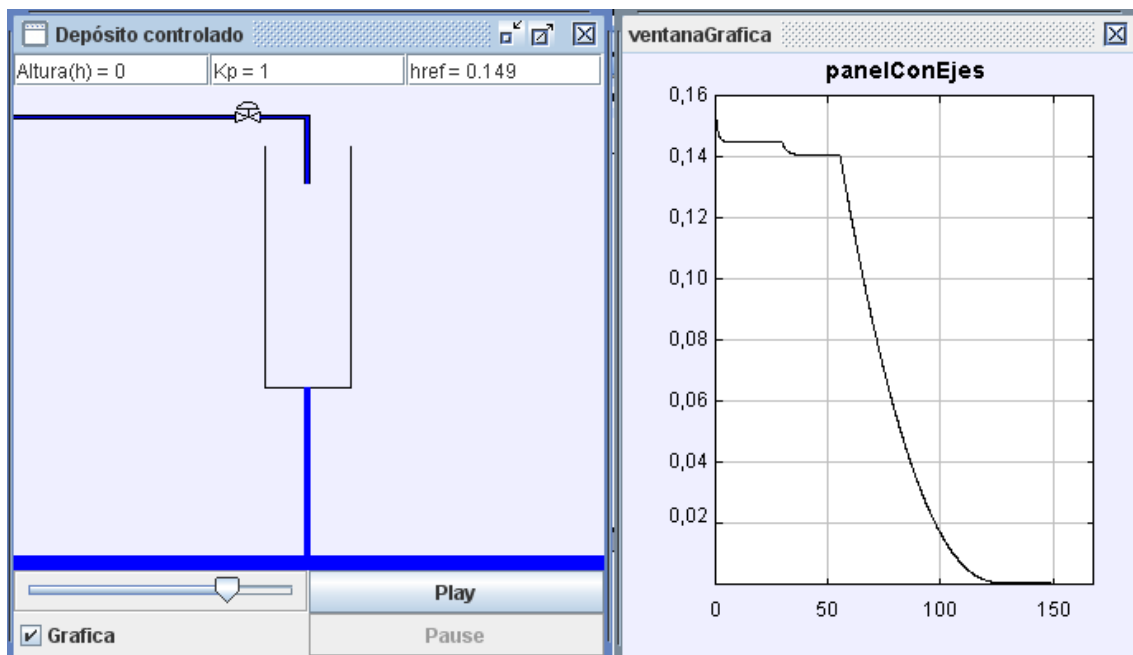


Figura 75: $h_{ref} = 0,149\text{m}$ y $K_p = 1$

Se observa como el sistema no es capaz de seguir a la entrada (h_{ref}), es decir comete un error muy grande, ese error es la diferencia de alturas, es decir, $error = 0.149 - 0.000 = 0.149m$, como se aprecia el error es igual a la entrada (h_{ref}), por tanto el sistema para una $K_p = 1$ no sigue la entrada y su comportamiento es peor que el del ejemplo anterior, por tanto, se observa que cuanto menor es K_p , mayor es el error. En este caso la variable de control u , es muy pequeña y no es capaz de seguir la entrada.

7.3.4 Ejercicio 4: $K_p = 1000$ y $h_{ref} = 0 - 0.2m$

Enunciado: fijar $K_p = 1000$, variar la referencia utilizando la barra deslizadora entre 0 y 0,2. ¿Es capaz de mantener la altura de referencia?

Como antes se ha dicho, en equilibrio la ecuación queda así:

$$0 = -\frac{a}{A}\sqrt{2gh} + \frac{k}{A} \cdot K_p \cdot (h_{ref} - h)$$

Así para una altura de referencia $h_{ref} = 0$ despejando, resulta $h = 0m$, la cual coincide con el valor obtenido en la simulación.

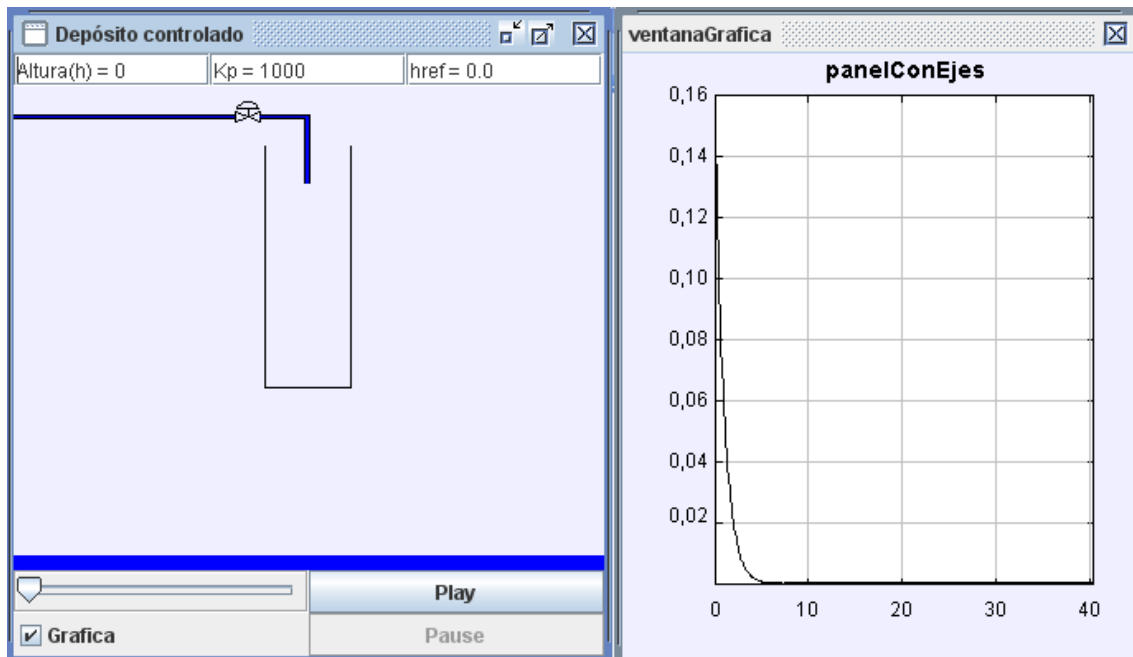


Figura 76: $h_{ref} = 0m$ y $K_p = 1000$

Para una altura de referencia $h_{ref} = 0.2m$ despejando, resulta $h = 0.195m$, con lo cual se ve que la salida sigue a la referencia salvo un pequeño error.

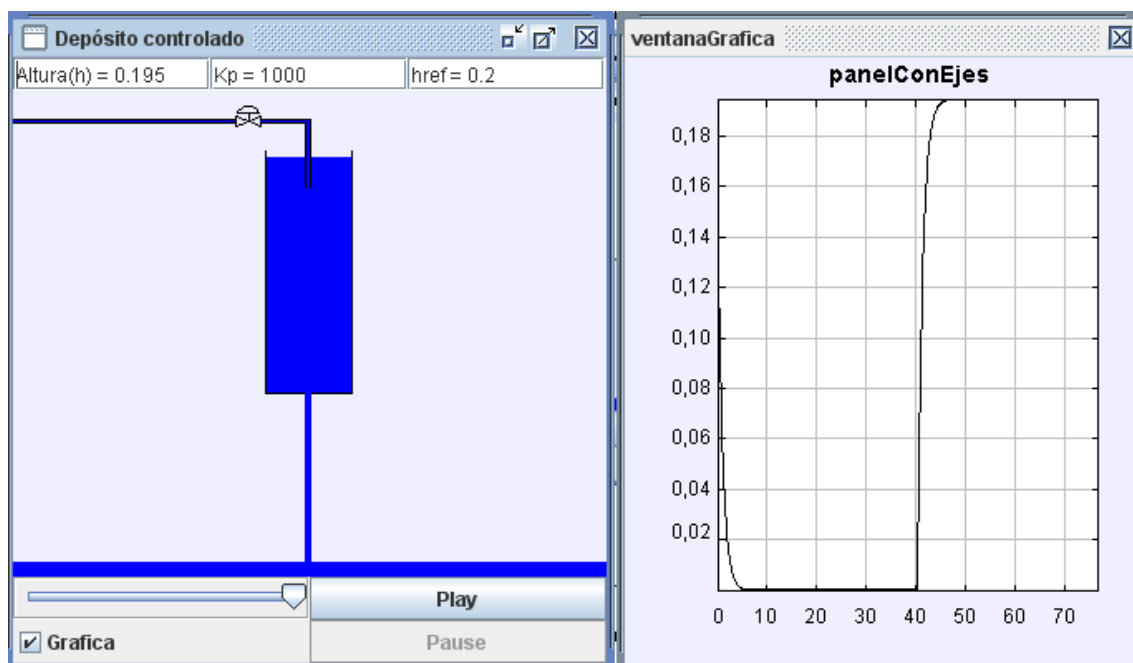


Figura 77: $h_{ref} = 0,2m$ y $K_p = 1000$

Se observa como el sistema es capaz de seguir a la entrada (h_{ref}). Cuanto menor es la altura mejor sigue a la entrada, para una entrada de 0m lo hace a la perfección, pero cuando se pone la altura máxima de 0.2m se ve como no llega a esos 0.2m sino que se queda en 0.195m, es decir comete un pequeño error, ese error es la diferencia de alturas, es decir, $error = 0.200 - 0.195 = 0.005m$, como se aprecia un error muy pequeño, por tanto el sistema para una $K_p = 1000$ sigue la entrada de forma correcta pero no exacta.

7.3.5 Ejercicio 5: $K_p = 10$ y $h_{ref} = 0 - 0.2$

Enunciado: fijar $K_p = 10$, variar la referencia utilizando la barra deslizadora entre 0 y 0,2. ¿Es capaz de mantener la altura de referencia?

Como antes se ha dicho, en equilibrio la ecuación queda así:

$$0 = -\frac{a}{A} \sqrt{2gh} + \frac{k}{A} \cdot K_p \cdot (h_{ref} - h)$$

Así para una altura de referencia $h_{ref} = 0m$ despejando, resulta $h = 0.0m$, la cual coincide con el valor obtenido en la simulación.

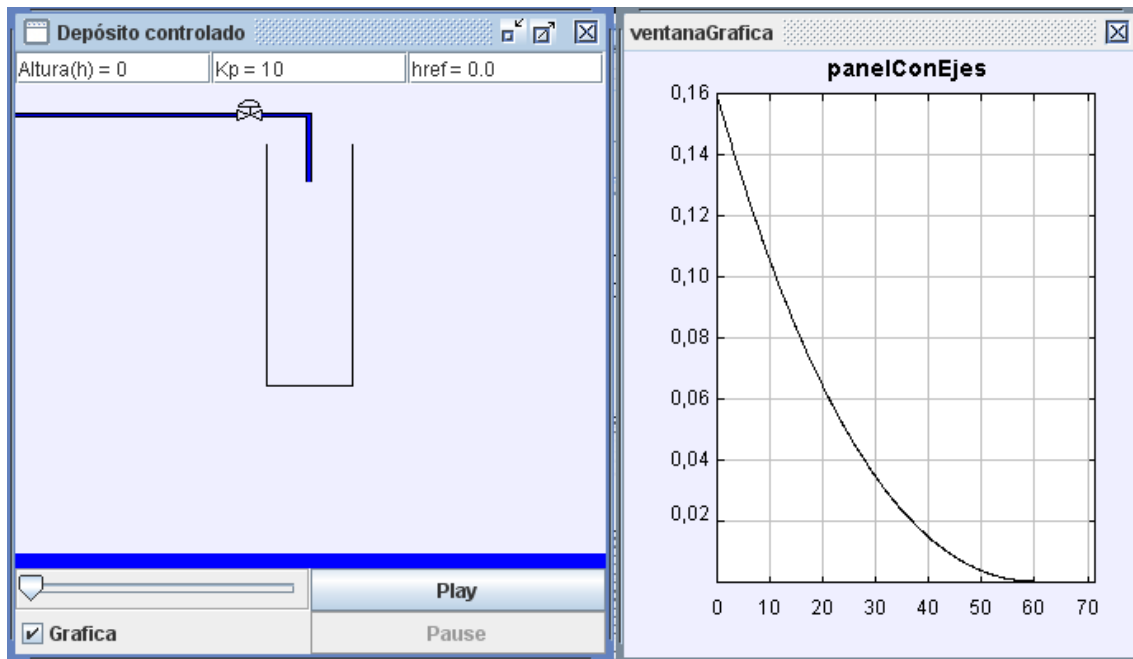


Figura 78: href = 0m y Kp = 10

Para una altura de referencia href = 0.2m despejando, resulta $h = 0.023\text{m}$, con lo que se ve que aunque la salida intenta seguir a la referencia, el error es muy grande.

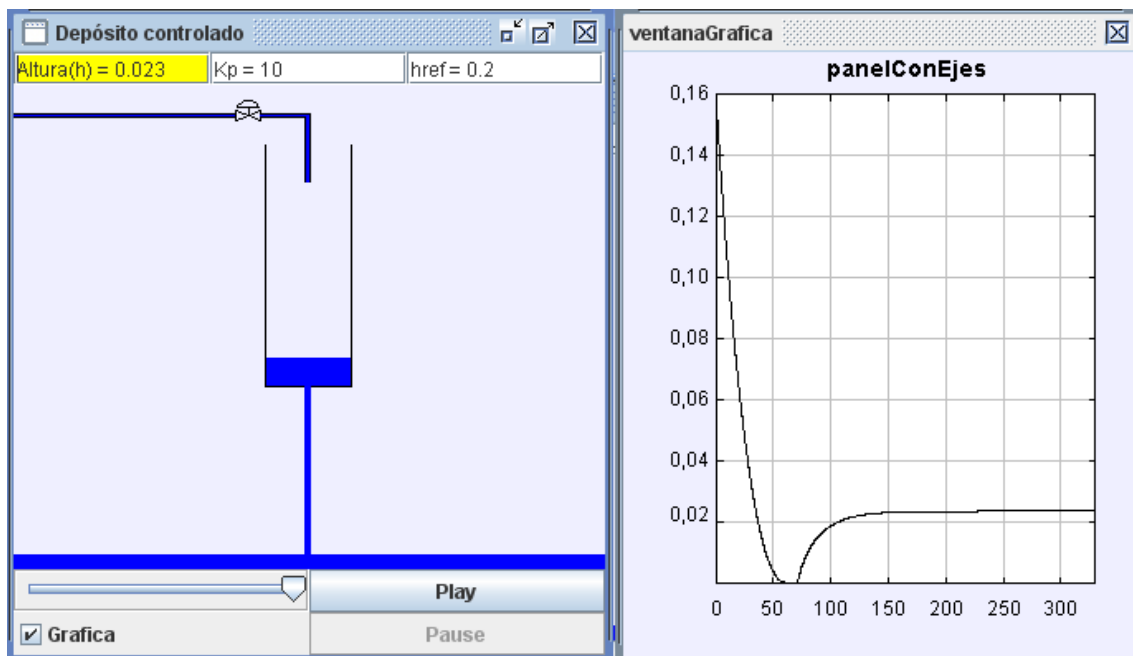


Figura 79: href = 0,2m y Kp = 10



Se observa como el sistema no es capaz de seguir a la entrada (h_{ref}), es decir comete un error muy grande, ese error es la diferencia de alturas, es decir, $error = 0.200 - 0.023 = 0.177m$. Como se aprecia el error es casi igual a la entrada, un error muy grande.

➤ Por tanto para valores de K_p pequeños el sistema se convierte en mucho más lento e imperfecto, ya que no sigue en ningún momento la entrada, esto ocurre porque la acción de control depende de K_p , y para valores de K_p pequeños u toma valores muy pequeños y no causa efecto en el sistema.



Capítulo 8

Conclusiones y trabajos futuros

8.1 Conclusiones

La implantación de laboratorios remotos y virtuales ofrece nuevas posibilidades en la educación al aumentar las oportunidades de experimentación. Tienen como ventajas los horarios de experimentación, ya que están disponibles a cualquier hora para cualquier persona, los alumnos acceden a estos laboratorios en cualquier momento con modelos simulados de los experimentos. Además de ampliar los horarios de experimentación, permiten que no sea necesario asistir a los laboratorios para realizar las prácticas, lo cual puede ser interesante para casos de personas que compaginan estudios con trabajo o alumnos con discapacidades.

Hay que destacar también, que su uso no aporta sólo capacidades sólo conocimientos de la propia asignatura, sino que además aportan a los alumnos conocimientos como son el uso de las TIC o el desarrollo de trabajos comunes entre los alumnos de manera virtual.

El uso de los laboratorios reales y virtuales posee muchas ventajas como ya se ha dicho, pero hay que añadir que en algunos casos puede requerir de los laboratorios reales para un correcto entendimiento del caso práctico (resolver problemáticas específicas de la interacción directa con los equipos como es la puesta a punto de los mismos, cableado, etc.), ya que los alumnos pueden aprender o entender conceptos acerca de los sistemas de forma virtual antes de llegar al laboratorio, lo cual que crea la ventaja de que el laboratorio presencial pueden aprovechar mejor el tiempo dedicado a la práctica.

En este proyecto, se ha conseguido, utilizar Easy Java Simulations para implementar prácticas para la asignatura de Señales y Sistemas. Se han desarrollado tres prácticas y se han presentado tres ejercicios a modo de ejemplo.



8.2 Trabajos futuros

Por último, siguiendo con el proceso de implantación de laboratorios remotos y virtuales, con este proyecto se pretende asentar la base de futuras herramientas del mismo tipo o posibles ampliaciones y mejoras de ésta misma.

Entre estas mejoras, podrían realizarse:

- Nuevas prácticas para la asignatura Señales y Sistemas.
- Prácticas para otras asignaturas del Departamento de Ingeniería de Sistemas y Automática.
- Dar el salto a laboratorios Remotos, intentando conectar el hardware con Easy Java Simulations, Matlab Server u otro tipo de herramientas que permitan al alumno hacer prácticas con los equipos de la Universidad desde sus casas.



Bibliografía



- [1] Dormido S. “Control Learning: Present and Future” *Annual Reviews in Control*, vol. 28. 2004
- [2] F. Esquembre. Creación de Simulaciones Interactivas en Java. Aplicación a la Enseñanza de la Física. Pearson Prentice Hall. Educación, 2005.
- [3] S. Dormido, J. Sánchez, H. Vargas, S. Dormido-Canto, R. Dormido, N. Duro, G. Farías, Ma. A. Canto. Dpto. Informática y Automática UNED. Laboratorios remotos y virtuales para la enseñanza.
- [4] Laboratorios virtuales en la educación. Por Arcadio de la Cruz Rodríguez, José Antonio Guerra García y Eduardo Lazarín Meyer.
División Preparatoria y Dirección de Innovación para la Academia Departamento de Ciencias Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) Campus Estado de México.
- [5] Laboratorios remotos en la Web. M. A. González, J. Adiego, L. F. Sanz, N. Bouab, J. Mass. Dpt. de Física, Escuela Técnica Superior de Ingenieros Industriales. Dpt. De Informática, Escuela Universitaria Politécnica, Universidad de Valladolid. España.
- [6] Nuevas aportaciones didácticas de los laboratorios virtuales y remotos en la enseñanza. L. Rosado, Universidad Nacional de Educación a Distancia (UNED).